Automatic Esophageal Abnormality Detection and Classification

UNIVERSITY OF
LINCOLN

Noha Ghatwary

School of Computer Science

College of Science

University of Lincoln

Submitted in partial satisfaction of the requirements for the
Degree of Doctor of Philosophy
in Computer Science

Supervisor Prof. Xujiong Ye

March 2020

Acknowledgements

First of all, I thank GOD for his showers of blessings throughout my Ph.D. journey to complete the research successfully.

I respectfully take this opportunity to acknowledge many people who deserve a special mention for their varied contributions during my Ph.D. research. I could never have achieved this work without their kind support and encouragement.

I would like to express my deepest sense of gratitude to my supervisor Prof. Xujiong Ye for being an outstanding advisor and excellent mentor. Her constant encouragement, patient guidance, support in various ways and invaluable suggestions made this work successful. It was a great privilege and honor to work and study under her guidance.

I would also like to express my sincere thanks to my second supervisor Dr. Massoud Zolgharni for his valuable support and guidance. I would also like to extend my gratitude to Dr. Faraz Janan, my third supervisor.

I would truly like to thank my mum and dad for their overwhelming affection and devotion that assisted me through every aspect of my life. Also, I express my thanks to my sister and brother for their continuous support, endless care, and encouragement and I would like to emphasize their role in my life and how much they have all endured for me to accomplish this work.

I wish to thank my husband for always encouraging me to finish this work and understanding how important it is to me.

I dedicate this work to each and every member of my family in an attempt to express my gratitude for their priceless sacrifices.

Abstract

Esophageal cancer is counted as one of the deadliest cancers worldwide ranking the sixth among all types of cancers. Early esophageal cancer typically causes no symptoms and mainly arises from overlooked/untreated premalignant abnormalities in the esophagus tube. Endoscopy is the main tool used for the detection of abnormalities, and the cell deformation stage is confirmed by taking biopsy samples. The process of detection and classification is considered challenging for several reasons such as; different types of abnormalities (including early cancer stages) can be located randomly throughout the esophagus tube, abnormal regions can have various sizes and appearances which makes it difficult to capture, and failure in discriminating between the columnar mucosa from the metaplastic epithelium. Although many studies have been conducted, it remains a challenging task and improving the accuracy of automatically classifying and detecting different esophageal abnormalities is an ongoing field. This thesis aims to develop novel automated methods for the detection and classification of the abnormal esophageal regions (precancerous and cancerous) from endoscopic images and videos.

In this thesis, firstly, the abnormality stage of the esophageal cell deformation is classified from confocal laser endomicroscopy (CLE) images. The CLE is an endoscopic tool that provides a digital pathology view of the esophagus cells. The classification is achieved by enhancing the internal features of the CLE image, using a novel enhancement filter that utilizes fractional integration and differentiation. Different imaging features including, Multi-Scale pyramid rotation LBP (MP-RLBP), gray level co-occurrence matrices (GLCM), fractal analysis, fuzzy LBP and maximally stable extremal regions (MSER), are calculated from the enhanced image to assure a robust classification result. The support vector machine (SVM) and random forest (RF) classifiers are employed to classify each image into its pathology stage.

Secondly, we propose an automatic detection method to locate abnormality regions from high definition white light (HD-WLE) endoscopic images. We first investigate the performance of different deep learning detection methods on our dataset. Then we propose an approach that combines hand-designed Gabor features with extracted convolutional neural network features that are used by the Faster R-CNN to detect abnormal regions. Moreover, to further improve the detection performance, we propose a novel two-input network named GFD-Faster RCNN. The proposed method generates a Gabor fractal image from the original endoscopic image using Gabor filters. Then features are learned separately from the endoscopic image and the generated Gabor fractal image using the densely connected convolutional network to detect abnormal esophageal regions.

Thirdly, we present a novel model to detect the abnormal regions from endoscopic videos. We design a 3D Sequential DenseConvLstm network to extract spatiotemporal features from the input videos that are utilized by a region proposal network and ROI pooling layer to detect abnormality regions in each frame throughout the video. Additionally, we suggest an FS-CRF post-processing method that incorporates the Conditional Random Field (CRF) on a frame-based level to recover missed abnormal regions in neighborhood frames within the same clip.

The methods are evaluated on four datasets: (1) CLE dataset used for the classification model, (2) Publicly available dataset named Kvasir, (3) MICCAI'15 Endovis challenge dataset, Both datasets (2) and (3) are used for the evaluation of detection model from endoscopic images. Finally, (4) Gastrointestinal Atlas dataset used for the evaluation of the video detection model. The experimental results demonstrate promising results of the different models and have outperformed the state-of-the-art methods.

Table of Contents

List of Figures

List of Tables

Acronyms

AUC area-under-the-curve

Avg. Pooling Average Pooling

BB Bounding Box

BE Barrett's Esophagus

BHI IEEE Journal of Biomedical and Health Informatics

BN Batch Normalization

CCV Color Coherence Vector

CII Contrast Improvement Index

CLE Confocal Laser Endomicrosciopy

CNN Convolutional Neural Networks

Conv Convolution Layer

ConvLstm Convolution Long Short Term Memory

DL Deep Learning

DWT Discrete Wavelet Transform

DYWT Dyadic Wavelet Transform

EAC Esophageal Adenocarcinoma

EC Esophageal Cancer

eCLE endoscopic CLE

FC Fully Connected Layer

FCN Fully Connected Neural Network

FLBP Fuzzy Local Binary Pattern

GERD Gastroesophageal Reflux Disease

GLCM Grey level co-occurrence matrix

GLRLM Grey level run-length matrix

GRU Gated Recurrent Unit

GT Ground Truth

HD-WLE High Definition WLE

HGD High Grade Dysplasia

HMM Hidden Markov Model

HOG Histogram of Oriented Gradients

IDA Iterative Deep Aggregation

IJCARS International Journal of Computer Assisted Radiology and Surgery

IoU Intersection over Union

JMI Journal of Medical Imaging

LBP Local Binary Pattern

LDF Local density function

LGD Low Grade Dysplasia

LOO-CV Leave-One-Out Cross-Validation

LOPO-CV Leave-One-Patient-Out Cross-Validation

LSTM Long Short Term Memory

mAP Mean of Average Precision

Max. Pooling Maximum Pooling

MICCAI International Conference on Medical Image Computing and Computer Assisted Intervention

MIUA Medical Imaging and Understanding Analysis

MLMI Machine Learning in Medical Imaging

MP-RLBP Multi-Scale Pyramid with Rotation Invariant LBP

MSER Maximally Stable Extremal Regions

NBI Narrow Band Imaging

OCT Optical Coherent tomography

OPF Optimum-Path Forest

pCLE probe CLE

RBF Radial Basis Function

R-CNN Region Based Convolutional Neural Network

ResNet residual networks

RF Random Forest

RNN Recurrent Neural Networks

SCC Squamous Cell Carcinoma

SIFT Scale-Invariant Feature Transform

SLIC Simpler Linear Interactive Clustering

SSD Single-Shot Multibox Detector

STD Standard Deviation

SURF Speeded Up Robust Features

SVM Support Vector Machine

Var Variance

VLE Volumetric Laser Endomicroscopy

WCE Wireless Capsule Endoscopy

WLE White Light Endoscopy

WT Wavelet Transform

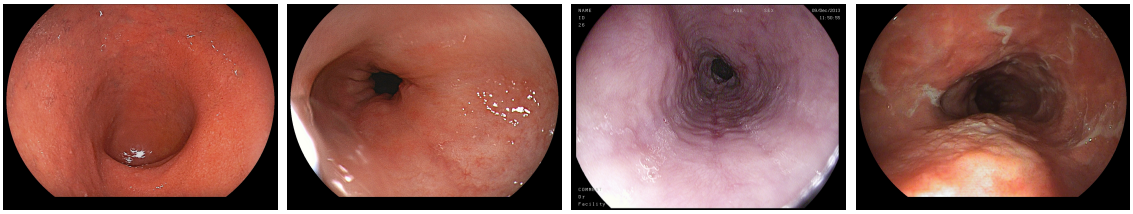
Chapter 1

Introduction

The incidence rate of Esophageal Cancer is rising dramatically in the last couple of years mainly due to late diagnosis and different risk factors. In 2019, the number of new cases reported in the United States was an average of 17,650 with 16,080 death cases. In this chapter, we provide an overview of esophageal abnormalities (precancerous and cancerous) and the importance of early diagnosis. Moreover, the problem statement, the motivation and the main objectives of the thesis will be introduced. A brief description of the thesis layout is given at the end of this chapter.

1.1 Overview and Problem Statement

Esophageal Cancer (EC) is an aggressive type of cancer that often remains asymptomatic until the late stages. It is the seventh most common cancer and the sixth leading cause of death from cancer in the world (*Worldwide cancer data* n.d.). The survival rate for EC patients varies from 4% to 40% depending on the development of the disease with a low survival rate of only 19% on a 5-year plan compared to other types of cancer such as: breast cancer (89%), lung cancer (55%) and stomach cancer (65%) (*Cancer Stat Facts: Esophageal Cancer* n.d.; Yousefi et al., 2018). Moreover, it is considered one of the main causes of the increased death rate in industrial countries, due to the difficulties of early detection and diagnosis. Different health factors can cause EC such as overweight and obesity, also, the increased consumption of tobacco and alcohol (Kamangar et al., 2009; Brooks et al., 2009).

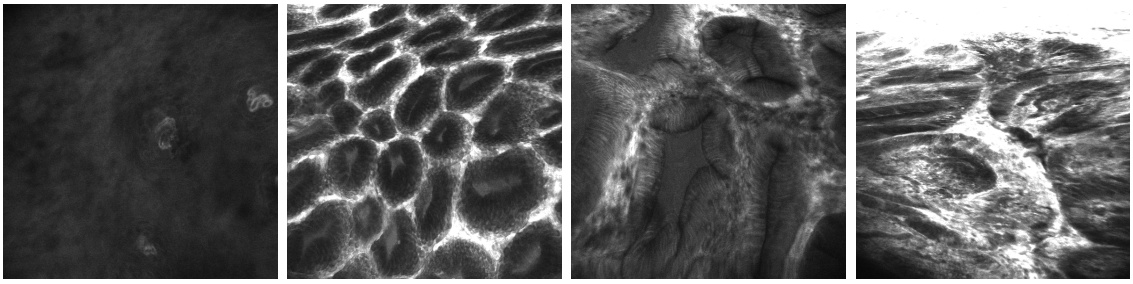


(a) (b) (c) (d)

Figure 1.1: Examples of different abnormal ties (precancerous and cancerous) from the esophagus captured by the endoscopic tool

EC occurs in the cells that line the surface of the esophagus and can appear anywhere along the esophagus tube. Early esophageal cancer typically causes no symptoms and mainly arises from untreated/unmonitored premalignant abnormalities (Shaheen and Ransohoff, 2002). Any inflammation or a small change in the cells of the esophagus tube is considered a precancerous stage such as Esophagitis and Barrett's Esophagus (BE). Different endoscopy tools can be used to examine the gastrointestinal tract (GI Tract) where the esophagus is located. Figure 1.1 represent different examples of endoscopic images showing examples of precancerous (Figs. 1.1a-1.1c) and cancerous (Fig. 1.1d) stages. The process of detection is done through an endoscopic examination while the grading of the cell deformation stages is confirmed by taking biopsy samples from the surface of the esophagus lining (Trovato et al., 2013). The different types of endoscopy modalities used in the examination process will be discussed in Chapter 2.

The process of detection has different challenges; the esophageal abnormal cells (precancerous and cancerous) can be located randomly throughout the esophagus tube (J. W. Cho, 2013). The abnormal region suspected of early cancer is very similar to the normal regions in the endoscope image (as shown in Fig. 1.1). Also, accurate detection requires a physician with significant experience as it is a difficult task to identify patterns associated with early cancer (Schölvinck et al., 2017). Moreover, studies show that early detection is often overlooked during endoscopy surveillance with a percentage of 20% to 25% (Kaise, 2015; Dik, Moons and Siersema, 2014; Visrodia et al., 2016). In addition to that, patients are required to have regular follow-ups through endoscopy examination to control the development of abnormalities. Generally, to increase survival rate, precancerous (i.e. Esophagitis and BE)



(a) (b) (c) (d)

Figure 1.2: Examples of different pathology grades captured by the CLE tool.

and early cancer stages need to be detected early to decrease the risk of the development into advanced stages (Bird-Lieberman and Fitzgerald, 2009; Sekiguchi and Oda, 2017).

Furthermore, as will be explained in Chapter 2, new technologies such as *Confocal Laser Endomicroscopy* (CLE) provide digital pathology images (i.e. to replace biopsy samples) that are instantly diagnosed by a physician. Examples of the digital pathology images provided by the CLE tool for different cell deformation stages are presented in Fig. 1.2. The investigation of these images showed that the deformation of cell stages is considered difficult to differentiate between them due to the very high similarity of the cell structure in each stage (Goldblum, 2003). It has been accounted especially by non-expert CLE endoscopists that there is an instability in the accuracy results when classifying the deformation stages, especially in early stages (Lim et al., 2011; Goetz, 2012). Also, it requires an observer that is very well trained with basic knowledge about histopathology to differentiate between normal and abnormal mucosa (Rajan et al., 2009).

For the aforementioned reasons, the computer-based detection and classification methods come to the aid of physicians for a more accurate diagnosis by acting as a second opinion. They can reduce the subjectivity of the physicians when performing diagnosis and eliminate the burden on patients during regular follow-ups. Additionally, it acts as a training method for junior physicians to learn how to identify abnormal regions and also practice how to use the endoscopic tools (i.e. such as CLE tool).

1.2 Motivation

Nowadays, the computer-based tools for medical diagnosis and treatment process is a fast-growing field. Computer-based automated systems can assist physicians as a second opinion in obtaining clinically significant information from endoscopes that are used to examine the state of an EC patient (Hiremath et al., 2003; J. d. Groof et al., 2019; Zhao et al., 2019). These systems can potentially detect and classify several abnormalities at different stages, increasing the chances of survival rate. Computer-based automated systems can have different phases as shown in Fig. 1.3. An endoscopy video/image passes through a preprocessing phase which enhances the images by noise reduction or feature enhancement. Afterward, the targeted abnormal area is automatically detected and in certain applications can also be segmented. Finally, the segmented/detected regions are automatically classified into the relevant cancer stage. In the literature, computer-based models can either perform only one operation (i.e. preprocessing, detection, segmentation or classification) or it can perform more than one operation.

Consequently, computer-based automated systems for esophageal abnormality detection and classification have started to grab attention with the increase of the EC incidents (L. A. d. Souza et al., 2018). There is a limited body of literature that addresses this issue which will be reviewed in the following chapters. The automatic esophageal abnormality classification and detection from endoscopic images have been reviewed by Domingues *et al.* (Domingues et al., 2019), Souza *et al.* (L. A. d. Souza et al., 2018) and Ghatwary *et. al.* (N. Ghatwary, A. Ahmed and Ye, 2017). Despite the efforts being dedicated to the esophageal abnormality problems, the process of detection and classification remains an ongoing research topic (Ebigbo, Palm et al., 2019).

Most of the existing studies on esophageal cell deformation stage classification are performed on CLE images, which captures a zoomed representation for cell structures providing a closer analysis. The main target of using the CLE is to classify the cell stage to replace the process of *Biopsy* which is required to confirm the diagnosis of the patient (Nakai et al., 2014). Biopsy samples may cause internal scars or bleed inside

Figure 1.3: Stages of Computer-based automated systems

the patient's organ when removing the tissue sample (Cequera and Leon Mendez, 2014). Moreover, the architecture of the tissue may be destroyed which might cause a limitation to the information gleaned from the sample. Therefore, decreasing the number of biopsy samples taken from the patient is needed to avoid affecting the patient's health which is possible by using the CLE.

The learning-based classification method requires understanding the internal structure of the cells at each stage properly. Most of the available classification methods are patch-based methods, where features are extracted from patches within the image which may lead to a decreased accuracy as the representation of cells and vessels is partitioned. Extracting features from the whole image representing the full structure of a CLE image can provide improved results. We intend to present a single-stage classification model that extracts features from the full CLE image after preprocessing enhancement stage to grade cell deformation.

Deep Learning (DL) has been tremendously useful in a wide range of different applications, such as computer vision, natural language processing, medical imaging analysis, and much more (Juefei-Xu, Naresh Boddeti and Savvides, 2017). Deep learning, specifically, Convolutional Neural Networks (CNN's), has become a conventional technique in medical image analysis (detection, classification, segmentation, etc...) (Litjens et al., 2017). Recent methods in the literature for esophageal abnormality detection have focused on using deep learning methods (Mendel et al., 2017). However, most of the CNN methods represented in the literature depend on *transfer learning* (i.e. learning the initial weights from a non-medical domain). Additionally, most of the methods investigate only one type of cancerous esophageal abnormality (more details will be represented in Chapter 4). We aim to provide a deep learning detection method that is trained end-to-end and considers the detection of different types of abnormalities from the endoscopic images and videos.

1.3 Aim and Objectives

This research aims to develop an automatic processing techniques to accurately detect and classify the abnormal region (precancerous and cancerous) in the esophagus tube from different endoscopic modalities. This thesis will focus on statistical learning-based medical image detection and classification techniques using hand-designed and machine-learned features. To achieve this, the objectives are:

- Developing and validating an automated **classification** method that can accurately classify the cell deformation stages in the esophagus tube from digital pathology **images** captured by the CLE tool.
- Building a **detection** model that can automatically locate abnormal regions (precancerous and cancerous) from High Definition While Light Endoscopy (HD-WLE) endoscopic **images** by exploring a new feature representation that combines hand-crafted features (Gabor Features) with machine-learned features (from designed DesneNet).
- Designing a framework that automatically **detects** esophageal abnormalities from endoscopic **videos**. This method will extract spatiotemporal information and include frame dependencies to improve the accuracy of detection throughout the video.
- Evaluating the proposed methods extensively by conducting experiments on different publicly available datasets. For grade classification, we validate on the *ISBI 2016* challenge dataset (*aidasub-clebarrett - Home* 2015), for abnormality detection from images we test on *MICCAI'15* challenge dataset (*Sub-Challenge Early Barrett's cancer detection* n.d.) and the open-access Kvasir (Pogorelov et al., 2017) dataset and finally for video detection we use the *GastroIntestinal Atlas* dataset (*El Salvador Gastrointestinal Atals* n.d.).

Throughout this research we managed to have access *first* to the CLE dataset for classification method provided by the Institute of Oncology at Padova (Italy) through the *ISBI 2016* challenge (*aidasub-clebarrett - Home* 2015) by participating in this challenge. Secondly, we succeeded in accessing the Early Barrett's Cancer detection

sub-challenge from MICCAI'15 (*Sub-Challenge Early Barrett's cancer detection* n.d.) that provides HD-WLE for abnormal cancerous regions. Later on, to further validate our detection models on precancerous stages, we obtained endoscopic images in esophageal precancerous stages from a publicly available dataset named Kvasir (Pogorelov et al., 2017) and the data were annotated by help from endoscopists. Finally, the video dataset was obtained from the open-access website GastroIntestinal Atlas (*El Salvador Gastrointestinal Atals* n.d.). The GastroIntestinal Atlas provides a large high-resolution video dataset for gastrointestinal endoscopy. We selected only the videos concerning the esophagus for our experiments.

Accordingly, in this thesis, we first implement a model to **classify** different grades of pathologic stages from CLE images. Followed by the **detection** of different abnormalities from selected endoscopic **images**. And, finally, we present the model for abnormality **detection** (precancerous and cancerous) from **videos**.

1.4 Contribution

This thesis describes a novel robust methods to detect and classify the precancerous and cancerous stages in the esophagus tube. The main contributions of this thesis can be summarized as follows:

- Developing a single-stage model that automatically classifies the esophagus cell deformation stages from CLE images (Chapter 3). The method enhances the internal image features using a novel enhancement filter that combines fractional integration with differentiation. Moreover, hand-selected features are extracted on multi-scale levels after studying the cell characteristics at each stage. The previous methods have suggested a multi-stage and patch-based classification method (Veronese et al., 2013) & (Grisan, Elisa Veronese et al., n.d.), while in this thesis we introduce a single classification model that extracts the features directly from a full image.
- Adapting different state-of-the-art deep learning object detection methods to successfully locate the abnormal regions from endoscopic images by extracting

the CNN features (Chapter 4). To the best of our knowledge, no work has been addressed before to comprehensively assess the performance of different CNN-based detection methods for detecting abnormal regions from esophageal endoscopic images.

- Proposing a unified framework to automatically detect both precancerous and cancerous regions from the endoscopic image by combing handcrafted features (Gabor features) with machine-learned features (CNN features) to enhance texture details for detection (Chapter 4). The Gabor filter responses calculated from endoscopic images are incorporated into the Faster R-CNN model while adopting a designed Densely Connected Convolutional Network (DenseNet) as the backbone network to extract CNN features. The previous methods based on CNN features (Van Riel et al., 2018) & (Mendel et al., 2017) mainly rely on transfer learning which means that the initial weights were learned from a non-medical domain while in our model we train the network end-to-end.

- Proposing a novel two-input network adapted from the Faster R-CNN to address the challenges of esophageal abnormality detection (Chapter 4). In this model, first, a Gabor Fractal (GF) image is generated using various Gabor filter responses considering different orientations and scales, obtained from the original endoscopic image that strengthens the fractal texture information within the image. Secondly, we incorporate DenseNet as the backbone network to extract features from both original endoscopic image and the generated GF image separately. Features extracted from the GF and endoscopic images are fused through bilinear fusion before ROI pooling stage in Faster R-CNN, providing a rich feature representation that boosts the performance of final detection.

- Proposing an efficient method to automatically detect different esophageal abnormalities from endoscopic videos (Chapter 5). We design a novel 3D Sequential Dense-ConvLstm backbone network that extracts spatiotemporal features from the endoscopic video. Our network incorporates 3D Convolutional Neural Network (3DCNN) and Convolutional Lstm (ConvLstm) to efficiently learn short and long term spatiotemporal features. We implement the network with

dense connectivity preserving the maximum flow of information between layers, therefore, the network is easily trained end-to-end. The generated feature map is utilized by a region proposal network and ROI pooling layer to produce a bounding box that detects abnormality regions in each frame throughout the video. Additionally, we investigate a post-processing method named Frame Search Conditional Random Field (FS-CRF) that improves the overall performance of the model.

The current research has resulted in **nine** papers (three peer-reviewed journals, five conference papers, and one journal under revision) that are listed in Appendix A. Also, I have received, **three** awards (1st place challenge award, Best Poster Presentation award and Best Paper award) that are listed in Appendix B.

1.5 Thesis Structure

The overall thesis layout is shown in Fig.1.4, where each chapter is summarized as follows:

Chapter 2 describes the clinical background of the esophageal abnormality classification and detection from endoscopes explaining the different stages of abnormalities from precancerous to cancerous stages. The focus will be on the different endoscopy modalities which are common for the examination of the esophagus. The datasets which are used for evaluation of each proposed method will be described, followed by the evaluation protocols for esophageal abnormality classification and detection.

Chapter 3 investigates a unified framework learning to classify esophageal abnormalities pathology stages using handcrafted features. Confocal Laser Endomicroscopy (CLE) is used to capture digital pathology images of the cell structure of the esophagus. An enhancement filter is proposed for preprocessing and different features are extracted from each image to classify the cell deformation stage. This chapter will also include the technical literature review on esophageal abnormality classification and analyze specifically the related research work which the most common CLE dataset within the field of esophageal classification

Chapter 4 presents different deep learning methodologies to automatically detect esophageal abnormalities from endoscopic selected images. The chapter introduces the combination of hand-designed and machine-learned feature for finding abnormal regions. The automatic detection of the abnormal region is further developed with a novel two-input network. Moreover, a general overview of CNN deep learning is presented in this chapter with related research work in the literature that uses the most common publicly available endoscopic dataset within the field of automatic esophageal abnormality detection.

Chapter 5 introduces a novel deep learning method to detect esophageal abnormalities from endoscopic videos. The deep learning method extracts the spatiotemporal features using a suggested backbone network to locate abnormal regions in different frames throughout the video. Additionally, a proposed post-processing method named Frame Search Conditional Random Field (FS-CRF) that improves the overall performance of the model by recovering the missing regions in neighborhood frames within the same clip is investigated.

Chapter 6 concludes and summarizes the thesis. Additionally, the chapter provides recommendations for future work.

Additionally, we provide a list of appendices that includes the following:

- Appendix **A**: List of Publications
- Appendix **B**: List of Awards
- Appendix **C**: Code Samples for Abnormality Grade Classification (Ch. 3)
- Appendix **D**: Code Samples for Abnormality Detection from Images (Ch. 4)
- Appendix **E**: Code Samples for Abnormality Detection from Videos (Ch. 5)

And, finally the list of references are presented.

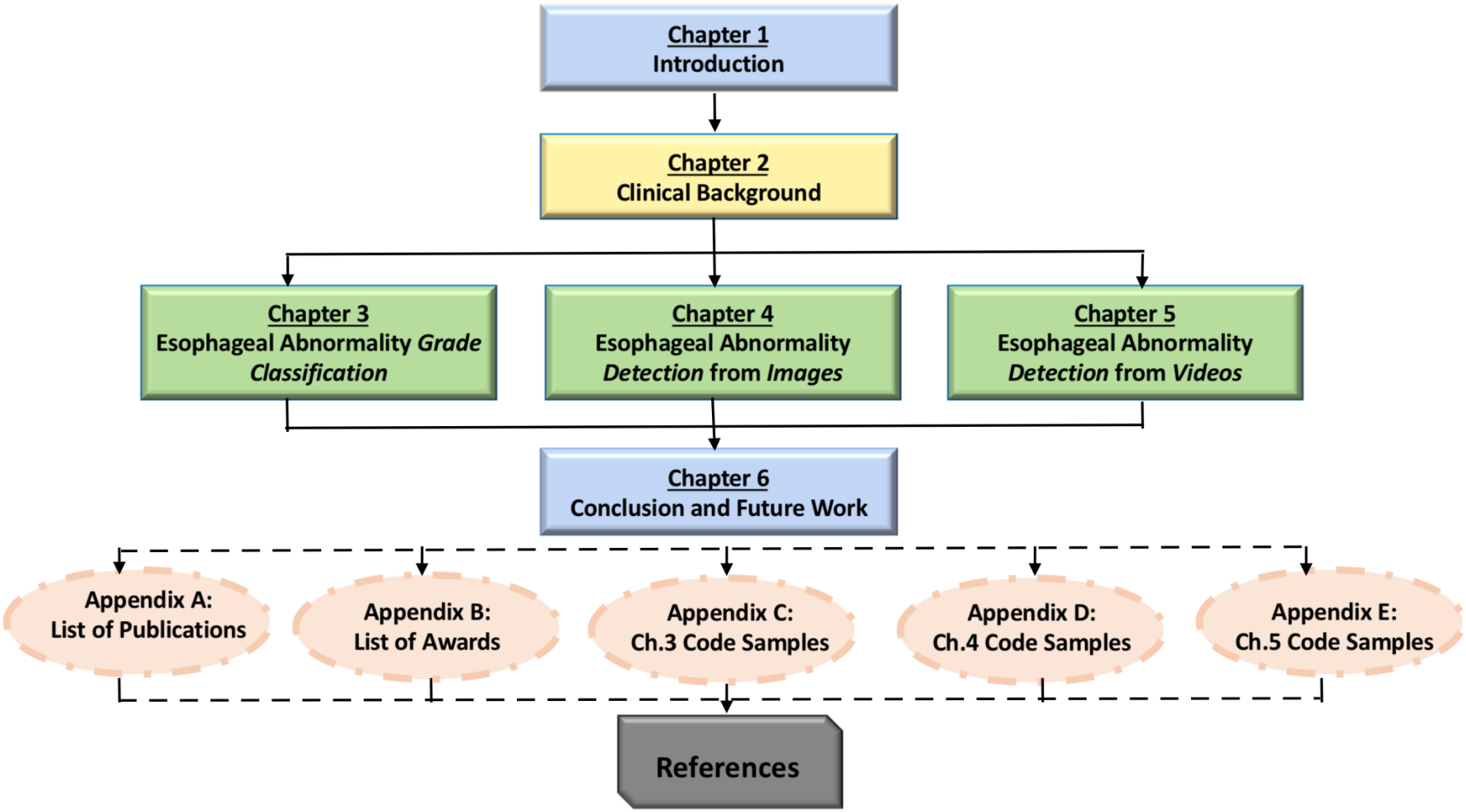


Figure 1.4: Pipeline Overview for the Thesis

Chapter 2

Clinical Background

2.1 Introduction

Application of medical imaging for the diagnosis of the esophageal abnormalities has developed over the past decades. The aim of esophageal examination through imaging is to locate the abnormal regions and classify their stages. This is beneficial for clinical operations such as diagnosis, treatment planning, surgical and radiotherapy preparation.

To examine the esophagus, endoscopic tools are used to view inside the patient's body and capture biopsy samples. There are different developments of endoscopes that can be used according to the required tasks. The examination using the endoscope has several advantages; it allows the physician to investigate the symptoms, diagnose the abnormal regions and finally allow surgical treatment.

In this chapter, we first explain the different types of esophageal abnormalities with its pathology stages. This is followed by a description of the different endoscopic tools and their application towards examining the esophagus. Moreover, the dataset used in this thesis is explained. Finally, the evaluation protocols are described in detail.

2.2 Esophagus Tube

The esophagus is a hollow muscular tube that connects the throat with the stomach as shown in Figure 2.1. The esophagus tube is located in front of the spine and

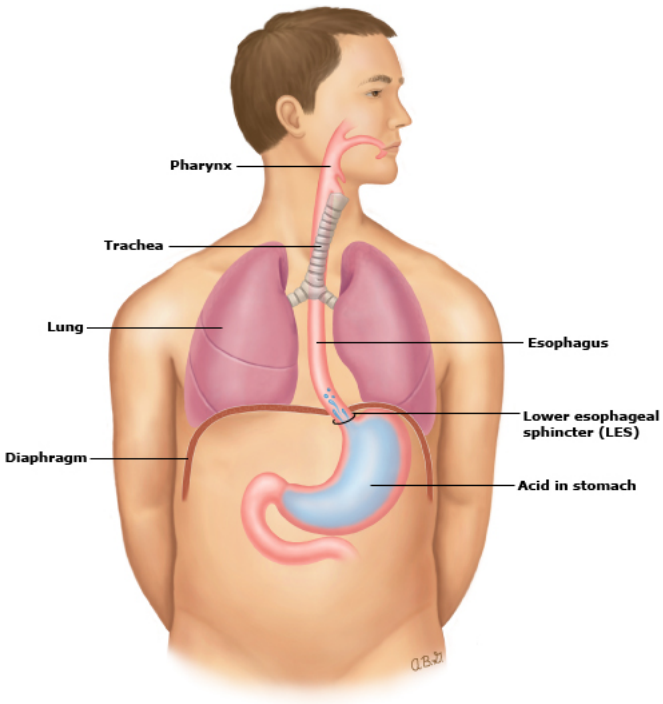


Figure 2.1: Illustration for the esophageal location inside a human body. The esophagus is the tube that connects between the pharynx (i.e. Throat) to the stomach. (*Can the lower esophageal sphincter be observed?* N.d.)

behind the lungs, trachea, and heart. Also, it passes through the diaphragm before entering the stomach. It is responsible for transferring the food and drinks from the mouth to the stomach. To move food to the stomach the muscles of the esophagus keep contracting while eating (i.e. this process is named peristalsis). Because of the food passing through it, the esophagus gets exposed to different materials with rough, soft and acidic textures. The average length of the esophagus is 25 cm long and is lined by pink mucosa tissues.

2.3 Esophagus Abnormalities

The abnormalities that appear in the esophagus tube can be divided into two categories: *precancerous* and *cancerous*. Any inflammation or a small change in the cells of the esophagus tube is considered as a precancerous stage such as ***Esophagitis*** and ***Barrett's Esophagus (BE)***. The untreated/unmonitored premalignant stages develop into esophageal cancer. Esophageal cancer usually occurs in the cells that

fill inside of the esophagus and can appear anywhere along the esophagus. There are two main types of esophageal cancer that are classified according to the type of cells (gland or squamous) known as: ***Esophageal Adenocarcinoma (EAC)*** and ***Squamous Cell Carcinoma (SCC)***. Early esophageal cancer typically causes no symptoms. Fig. 2.2 illustrates examples from endoscopic images capturing different types of abnormalities. Each of these four abnormalities will be briefly described in the following subsections.

2.3.1 Esophagitis

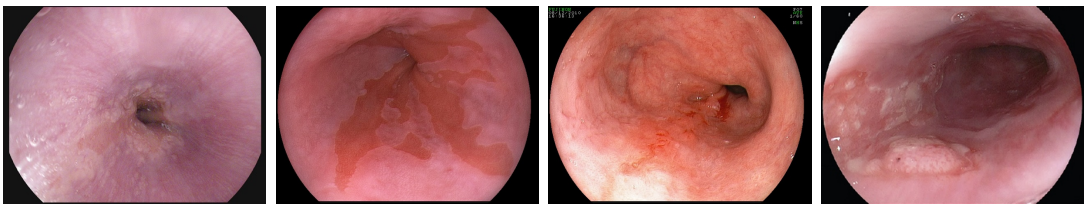
Esophagitis is an inflammation or infection of the lining of the esophagus. The esophagitis can be an outcome of radiation treatment or flowback of gastric acids such as reflux (i.e. known as Gastroesophageal Reflux Disease (GERD)), vomiting and occurrence of a hernia. The detection of esophagitis is important to start early treatment in order to eliminate the pain and reduce the possibility of further complications. Fig. (2.2a) illustrates an example of an endoscopic view of the Esophagitis.

2.3.2 Barrett's Esophagus (BE)

BE is the deformation of the healthy cells above the lower esophageal sphincter. It starts to appear when the normal squamous epithelium is replaced by metaplastic mucosa epithelium containing gastric or intestinal mucosa (Coleman et al., 2014) (i.e. can evolve from non-treated esophagitis and GERD). BE is considered the main precancerous condition that has a high risk to turn into esophageal cancer (Rajendra and Sharma, 2017; Flejou, 2005). An example of BE view is shown in Fig. (2.2b).

2.3.3 Esophageal Adenocarcinoma (EAC)

EAC appear in the gland cell of the esophagus tube. Glandular cells in the lining of the esophagus produce and release fluids such as mucus. It most often occurs in the lower part of the esophagus tube (near to the stomach). A patient that has BE are



(a) Esophagitis (b) BE (c) EAC (d) SCC

Figure 2.2: Example of the endoscopic view for the four different abnormality types: Example of the endoscopic view for the four different abnormality types: (a) Esophagitis, (b) BE, (c) EAC, (d)SCC

at increased risk to have EAC, as it is considered the most common precancerous stage the develops into cancer. EAC endoscopic view is shown in Fig. (2.2c).

2.3.4 Squamous Cell Carcinoma (SCC)

SCC appear in the squamous cell of the esophagus tube. Squamous cells are the thin flat cells that line the esophagus surface. SCC can be found anywhere along the esophagus tube but it is most often located in the upper and middle part. Fig. (2.2d) shows an example of an endoscopic view of the SCC

2.3.5 Pathology Stages of Esophagus Abnormalities

According to Mainz Confocal Barrett's Classification (Kiesslich, Gossner et al., 2006), the transformation of the cells in the esophagus tube has a different vessel appearance and cell structure. The stages can be categorized into four histopathology grades; ***Normal Squamous*** **(NS)** is the normal stage where the patients have no disease, ***Gastric Metaplasia (GM)*** is the first stage of cell deformation accompanied with mucus, ***Intestinal Metaplasia (IM)*** is the main precancerous stage **-**often considered as proper Barrett's Esophagus**-** with dysplasia in the esophageal path (K. K. Wang and Sampliner, 2008), patients who have GM can also have IM (Veronese et al., 2013) and finally ***Neoplasia Mucosa (NPL)*** is the later stage that might be cancerous. Each of these stages has a special appearance, the vessels of GM have a regular shape that appears in deeper parts of the mucosal layer and the cells have a regular shape with a cobblestone appearance. In the case of IM, the epithelium starts to be visible in the upper and the deeper part of the mucosal layer

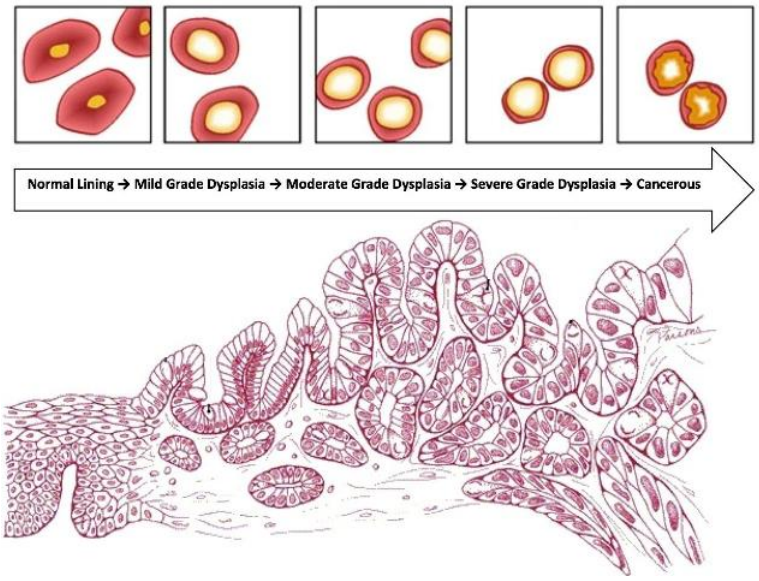


Figure 2.3: Cell transformation stages from normal to dysplasia (mild, moderate and severe) to cancer in esophagus lining (*Johns Hopkins Department of Pathology: Barrett's Esophagus* n.d.).

accompanied with goblet and cylinder dark cells. When reaching the NPL stage the vessels and cells have an irregular appearance with dark contrast (Watson, 2014). Fig. 2.3 demonstrates the transformation stages from normal cells until reaching the cancerous stage.

2.4 Endoscopy Tools

Endoscopy is a non-surgical process that examines the different cavities within a human body (Gotoda, 2007). There exist several types of endoscopy procedures in the medical field based on the examined area such as Colonoscopy (colon), Thoracoscopy (lungs), Neuroendoscopy (brain & spine), etc. For the examination of the Upper Gastrointestinal Tract (GI Tract) where the esophagus is located, the procedure is called Esophagoscope or Gastroscopy (Liedlgruber and Uhl, 2011). During the esophagus examination, the doctor passes the endoscope (i.e. which is a flexible tube with a camera and light attached to it) through the mouth into the esophagus allowing the doctor to view the esophagus on a TV monitor as shown in Fig. 2.4. Several endoscopic technologies are developed for examining different areas in the GI. In this section, we will focus on the endoscopic tools used to obtain datasets

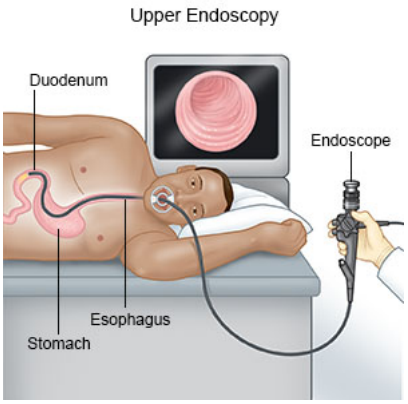


Figure 2.4: The process of esophgus examination using the endospce tool and viewing internal cavity on TV monitor (*UPPER ENDOSCOPY* n.d.).

used in this thesis such as White Light Endoscopy (WLE), High-Definition White Light Endoscopy (HD-WLE) and Confocal Laser Endomicroscopy (CLE). Moreover, we give a brief description of other endoscopic tools used for esophagus examination and have been used in other studies available in Literature.

- **White Light Endoscopy (WLE) - High Definition WLE (HD-WLE)**

  The standard WLE and HD-WLE are the primary tools used for examination to detect esophageal abnormalities (Behrens et al., 2011). WLE enables the image to be zoomed in up to 850,000 pixels (Haringsma et al., 2001) and uses the reflection of white to form an accurate to life representation of the mucosa. Although the WLE does not exploit the full range of visual difference between normal and neoplastic tissue, it is still used frequently (Gill and Singh, 2012). Nowadays, the HD-WLE have widely replaced the WLE in most of endoscopic units. The HD-WLE magnifies the image 115 times using optical magnifier producing an image with an image resolution of more than one million pixels (Naveed and Dunbar, 2016). The HD-WLE provided the endoscpists the ability to examine and visualize the mucosal abnormalities (Kwon et al., 2009). These two endoscopes are very effective in detecting abnormalities but they cannot differentiate between the epithelium type, therefore, random biopsy samples must be taken to confirm the diagnosis (Kara et al., 2005). ***In this thesis, we use images captured by the WLE and HD-WLE to evaluate our proposed automatic detection methods fro images and videos.***

- **Confocal Laser Endomicrosciopy (CLE)**

  CLE is regarded as one of the latest technologies used for the examination of cell and subcellular imaging up to 250 micrometers below the mucosal surface (Kiesslich, Goetz et al., 2005). It is a real-time endoscopic tool that allows both *imaging* and *patahlogoy diagnosis*(Buchner and Wallace, 2015). A blue-colored laser is focused on a single point in a microscopic field of view overlapping the optical path so the point of illumination matches to the point of interest within the specimen (Beg, A. Wilson and Ragunath, 2016). The blue laser light is used to focus on the mucosa while injecting a contrast agent called intravenous. The reflected light is filtered through a pinhole, therefore, decreasing light scatter, producing highly detailed images from a thin focal plane (East et al., 2016). There are two types of CLE-based systems that are used in routine clinical practice and research (Julia Liu, Dlugosz and Neumann, 2013; De Palma, 2009). Firstly, an endoscopic CLE (eCLE) (Pentax, Tokyo, Japan) (Wallace and Fockens, 2009), a confocal scanner has been integrated into the distal tip of a flexible endoscope. The eCLE captures 0.8 frames per second at a resolution of 1024x1024 (Becker et al., 2008). Secondly, a probe CLE (pCLE) (Cellvizio Endomicroscopy System; Mauna Kea Technologies, Paris, France) that is composed of a flexible mini probe, which is introduced through the working channel of a standard endoscope (T. D. Wang et al., 2007). The pCLE captures 12 frames per second but with a much lower resolution and smaller field of view compared to the eCLE. It was reported that an appropriate diagnosis of the histology grade using CLE might need a less number of biopsy samples taken from the patient. It is also considered an important field that will grab attention for more research in the field of automatic classification (Liedlgruber and Uhl, 2011). ***In this thesis, we use images captured by the eCLE to evaluate our proposed automatic classification method.***

- **Wireless Capsule Endoscopy (WCE)**

  WCE is a non-invasive technology that exists in a pill shape which can approximately capture an average of 50,000 images throughout an examination

of 7-8 hours where these images are sent and oriented remotely throughout the examination (Ramirez et al., 2005). The problem with WCE is that its position cannot be controlled which might lead to uncertainty in its diagnosis. It is usually used to examine the colon and small intestine.

- **Chromoendoscopy**

  Chromoendoscopy is an endoscope that injects a Methylene Blue (a type of dye) to select the stained segments for the biopsy process. The process seems simple, but it requires an increased examination time. In addition to that, the pattern classification process showed instability and the experts required more training to be specialized in categorizing it. Moreover, the Methylene Blue has a risk of causing carcinogenesis and damage to the DNA (Olliver et al., 2003).

- **Narrow Band Imaging (NBI)**

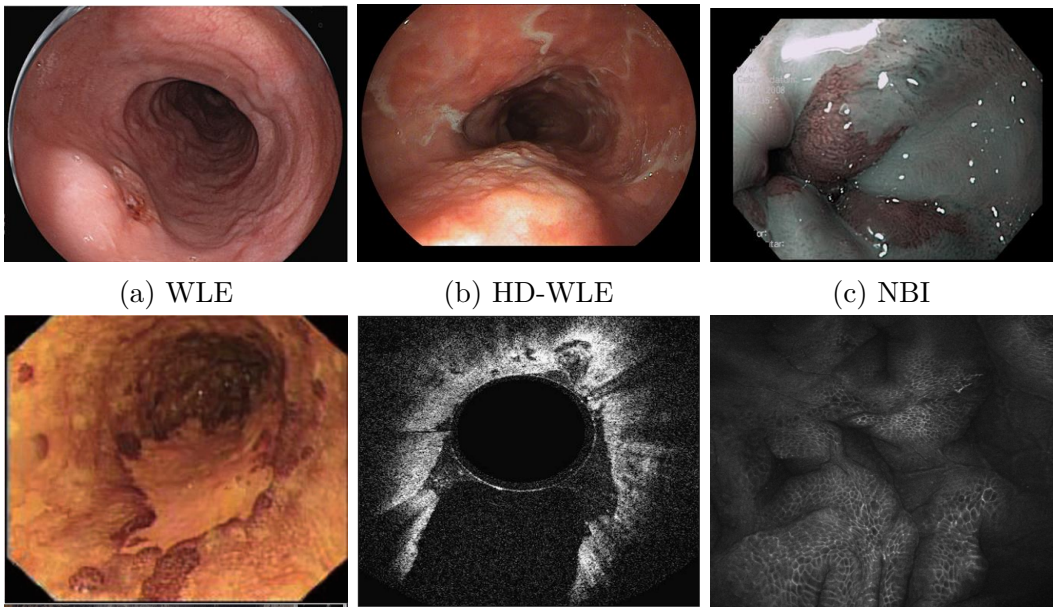
  NBI can study the vascular pattern and mucosal by enhancing its surface resolution. NBI utilizes a short wavelength of blue light that is supposed to absorbed by hemoglobin in the blood. Some studies showed NBI was superior to the WLE in detecting the HGD while other studies showed its disability in detecting neoplasia (Singh and Yeap, 2015).

- **Optical Coherent tomography (OCT)**

  OCT uses light waves to capture the scattered coherent light for mucous. It has the ability to find the transformation of early dysplasia. Studies showed that OCT has an accuracy of 61% to 76% detection from endoscopists (Qi et al., 2010). OCT is recognized to be effective, although it is not commonly used nowadays (Shahid and Wallace, 2010).

- **Volumetric Laser Endomicroscopy (VLE)**

  VLE is a second-generation endoscopic technology of the OCT that is a balloon-based imaging modality. It forms a two-dimensional image by using the optical scattered difference of the tissue decomposition. It produces cross-sectional images of tissues with an axial resolution of up to 10 micrometers. It scans



(a) WLE (b) HD-WLE (c) NBI

(d) Chromoendoscopy (e) OCT (f) CLE

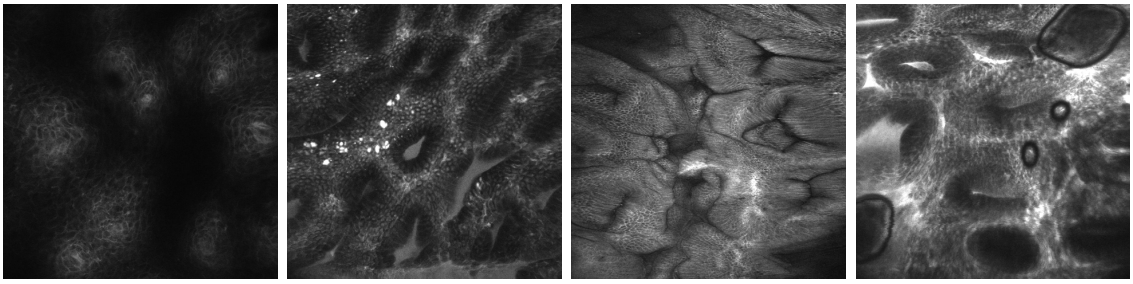
Figure 2.5: Examples of esophageal abnormalities captured with different endoscopic tools, (a) WLE, (b) HD-WLE, (c) NBI, (d) Chromoendoscopy, (e) OCT and (f) CLE.

surface and subsurface tissues for signs of abnormality in a high-speed allowing the diagnosis to happen in a real-time manner.

Each of the described endoscopes is used for a specific reason. The WLE and HD-WLE are intended to detect abnormal areas, while NBI and Chromoendoscopy are more suitable for tissue characterization. Moreover, the CLE is used for histological confirmation (M. H. Lee et al., 2012). Fig. 2.5 represents samples of images captured by the endoscopes during the esophagus examination.

2.5 Datasets Used in the Thesis

Four datasets are used in this study to develop the algorithm and establish a comprehensive evaluation and comparison with other methods in the literature. The first dataset is used for the pathology stage classification, the second and third datasets are used for detection from endoscopic images and the fourth dataset is used for detection evaluation from endoscopic videos. The following subsections will describe the details of data acquisition, data properties and annotation for each dataset.



(a) NS (b) GM (c) IM (d) NPL

Figure 2.6: Examples from the CLE dataset showing images from the four pathological stages: (a) NS, (b) GM, (c) IM and (d) NPL.

2.5.1 CLE dataset

A CLE dataset consisting of 557 images of 4 different histopathology grades from 96 patients were used to test the efficiency of the proposed model (**NS** 12 patients of 45 images, **GM** 10 patients of 41 images, **IM** 58 patients of 402 images and **NPL** 16 patients of 68 images). Endomicroscopy were performed by two experienced endoscopists at the European Oncological Institute (IEO, Milan, Italy) and Veneto Institute of Oncology (IOV, Padova, Italy) during routine clinical surveillance endoscopy in patients with BE, using a confocal laser endoscope (EC-3870CIFK; Pentax, Tokyo, Japan), allowing simultaneous endoscopy and endomicroscopy. The preparation of the patients includes conscious sedation. The confocal images were obtained after injection of 10% fluorescein sodium. The resolution of each image is 1024×1024 (corresponding to 500×500 µm) that was obtained at a scan rate of 0.8 frames per second using an optical slice thickness of 7 µm and stored digitally. The range of the z-axis was 0-250 µm below the surface layer.

2.5.2 MICCAI ENDOVIS'15 Dataset

The dataset of the sub-challenge Early Barrett Cancer detection from *EndoVis MICCAI 2015* challenge (*Sub-Challenge Early Barrett's cancer detection* n.d.) is composed of a total of 100 HD-WLE images with a resolution of (1600×1200) and gathered from 39 patients. The images are divided into 50 images without any cancer signs obtained from 17 patients and the other 50 with cancerous regions from 22 patients diagnosed with esophageal adenocarcinoma (EAC). Lesions found in the

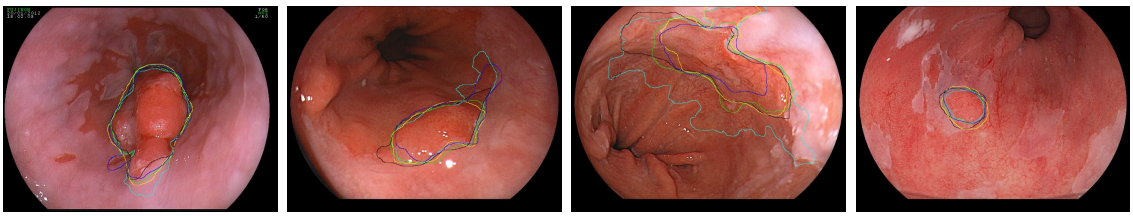


Figure 2.7: Examples from the Miccai dataset showing images with EAC with the annotation by the experts.

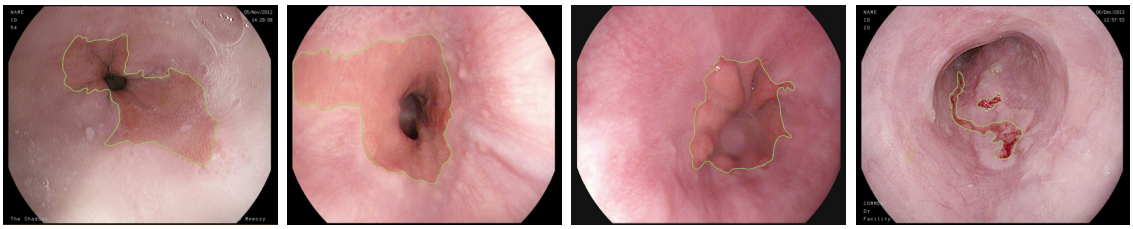


Figure 2.8: Examples from the Kvasir dataset showing images with Esophagitis abnormalities with the annotation by an expert.

abnormal images have been annotated by five leading experts in the field to obtain gold standard as shown in Fig. 2.7.

2.5.3 Kvasir Dataset

The Kvisar Dataset (Pogorelov et al., 2017) is an open-access dataset that provides classified sets of images inside the gastrointestinal (GI) tract showing anatomical landmarks such as (*Z-line, pylorus* & *cecum*), pathological findings such as (*esophagitis, polyps* & *ulcerative colitis*) and images related to the removal of lesions such as (*dyed resection margins* & *lifted polyp*). For our evaluation of the detection model, we only used the **Esophagitis** dataset that is composed of 1000 images obtained from different patients with a resolution that varies from 720×576 to 1920×1072. An expert in the field has manually annotated abnormalities in the images. Fig. 2.8 illustrates samples from the Kvasir dataset with the annotation by the expert.

2.5.4 Gastrointestinal Videos Dataset

The dataset of videos is from the online open-access website *GastroIntestinal Atlas* (*El Salvador Gastrointestinal Atals* n.d.). The dataset includes 42 endoscopic videos

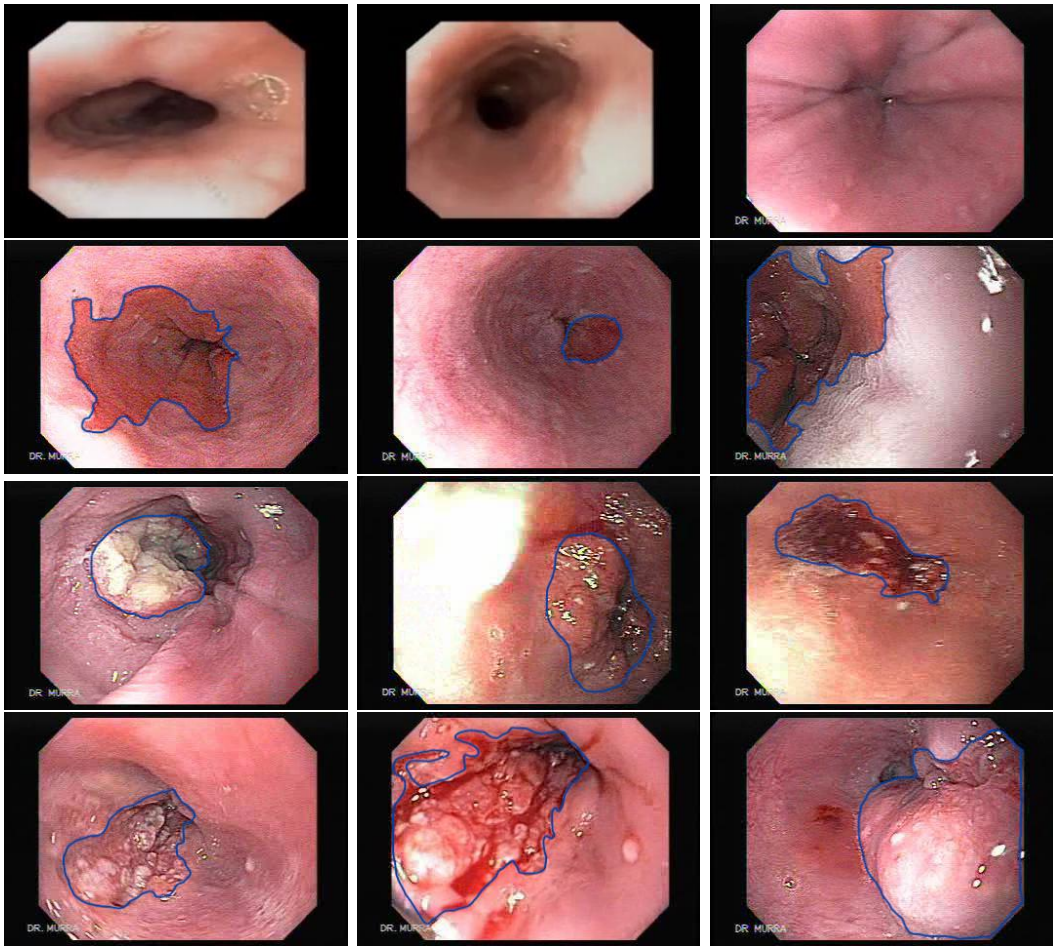


Figure 2.9: Examples of frames from the video dataset used in the evaluation of the proposed model. The first row shows samples from normal video frames. The second row illustrates samples from precancerous BE videos. Finally, third & forth represents cancerous samples from EAC and SCC videos respectively. The annotation by the expert is shown in blue for both the BE, EAC and SCC frames.

(total of 42,425 frames) gathered from 16 patients with different types of abnormalities. Each video has an average duration of 50 seconds (The time ranges from 30 seconds to 4 minutes per video). Additionally, the video frames have a resolution of 240×352 with a frame rate of 30 frames per second (fps) and divided into three categories; normal, precancerous and cancerous. The 42 videos are classified as follows: *Normal* (1 video from 1 patient), *BE* (24 videos from 8 patients), *EAC* (9 videos from 3 patients) and *SCC* (10 videos from 3 patients). The abnormality regions in the dataset were annotated by experts in the field. Samples of frames from the video dataset with the annotations are shown in Fig. 2.9.

2.6 Evaluation Protocols

As explained in the previous section; each dataset described is used for the evaluation for each phase separately. For the **classification** phase; the CLE dataset is evaluated using multi-label classification to classify the images into four categories: *NS, GM, IM* & *NPL*. However; the evaluation approach in the field of esophageal abnormality **detection** is comparing the detected area to the ground-truth area (which are provided by the expert).

2.6.1 Evaluation of Classification

- True Positive (TP): Indicates the number of correct prediction of target class.
- True Negative (TN): Indicates the number of correct prediction of the other class.
- False Positive (FP): Indicates the number of incorrect prediction of the other class classified as target class.
- False Negative (FN): Indicates the number of incorrect prediction of the target class classified as other class classified class.

Table 2.1 illustrates the confusion matrix for the above-mentioned classification categories.

Table 2.1: Confusion matrix for classification category

		True condition	
		Postive Condition	Negative Condition
Predicted Condition	Postive Prediction	TP	FN
	Negative Prediction	FP	TN

We employ the standard performance metrics generally adopted in medical imaging classification methods; *Accuracy, Sensitivity (i.e. also know as Recall), Specificity, Precision*, and *F-measure*. These measures are calculated by using the TP, TN, FP & FN defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.3}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$F - Measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{2.5}$$

2.6.2 Evaluation of Detection

The proposed detection methods generate bounding boxes to identify the detected region. In the literature, there exist two evaluation measures used to compare the predicted bounding-box (detection output area) with the annotated region (ground-truth area) known as **Intersection over Union (IoU)** and **Dice Similarity**. The *IoU* (i.e. also known as Jaccard index) measures the overlap area between the predicted detection and the ground-truth divided by the area of **union** between the predicted area and ground-truth. On the other hand, the *Dice Similarity* measures the overlap area between the predicted detection and the ground-truth divided by the **sum** area of both regions (i.e. total number of pixels). The IoU is commonly used in the literature to evaluate detection methods while the Dice Similarity is mostly used to evaluate segmentation methods (Kong et al., 2016; Cai and Vasconcelos, 2018; Tychsen-Smith and Petersson, 2018; Pereira et al., 2016). Therefore, in this thesis, we employ the IoU for the evaluation of the detection performance.

We use the same measures described in eq.(2.1) to (2.5). The detection bounding box is recognized as a TP if it overlaps with **IoU** value at a certain threshold and FN otherwise. The IoU is defined as:

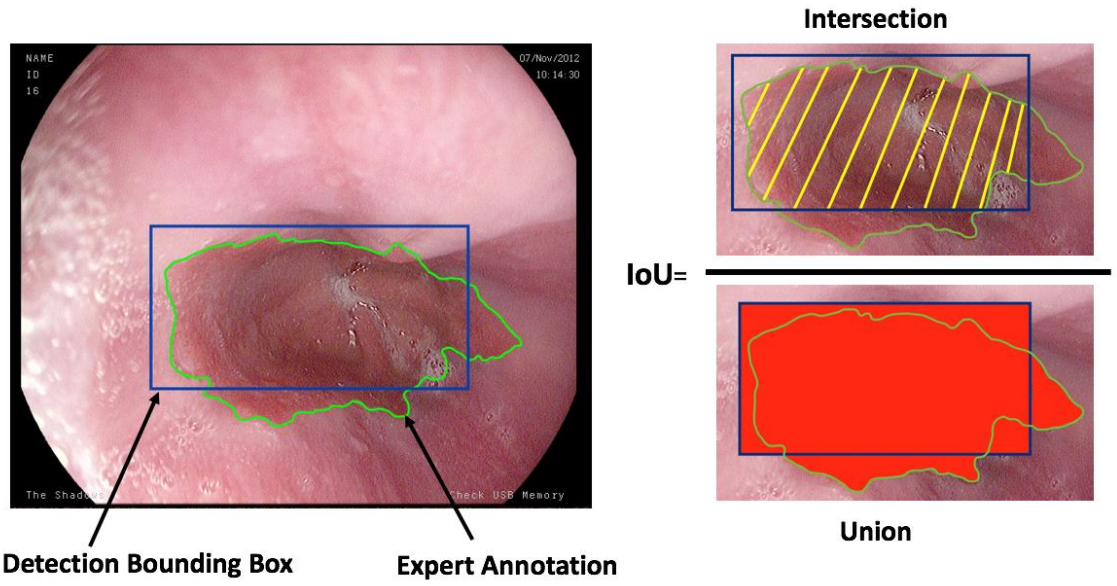


Figure 2.10: Illustration of the regions which are used for evaluation of the detection

$$IoU = \frac{A_{gt} \cap A_p}{A_{gt} \cup A_p} \tag{2.6}$$

Where, A_{gt} is the area of the ground-truth of experts annotation and A_p is the predicted bounding box from the detection method. Fig. 2.10 shows an example of the evaluation of a detected region with the generated bounding box and compared to the annotation.

Additionally we include the Mean of Average Precision (mAP) to evaluate the performance of detection localization by the proposed methods. The mAP measures the mean of Average Precision (AP) of the detection output; the AP measures the precision at different recall intervals defined as:

$$AP = \frac{1}{11} \sum_{recall_i} Precision(Recall_i) \tag{2.7}$$

Furthermore to investigate the differences in the results of recall, precision and F-measure between the different detection models (i.e. abnormality detection from images and videos), the **two tailed paired t-test** at a 95% confidence level was performed on the different datasets. The two-tailed t-test is more effective than

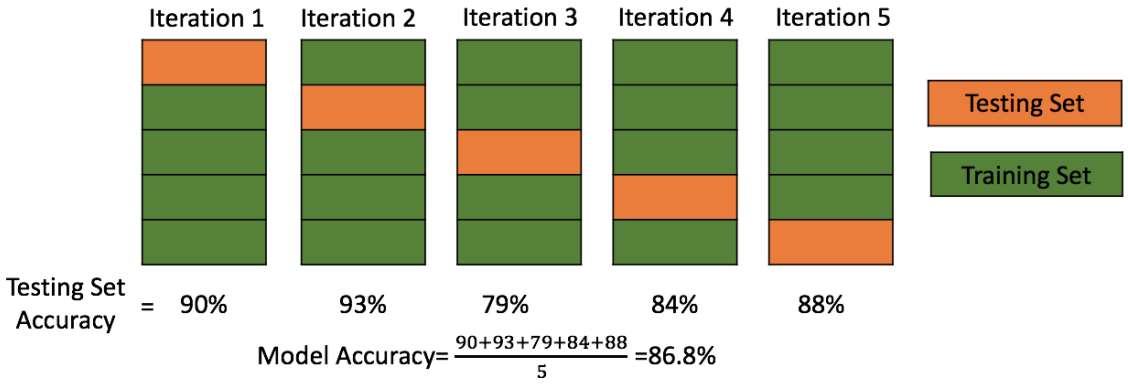


Figure 2.11: Example of N fold Cross-Validation operation using $N = 5$.

the single-tailed when analyzing the results as it requests a large difference to conclude the significant difference. A two-tailed test represents the differences between the groups you are comparing where it tests the possibility of positive or negative differences.

2.6.3 Cross Validation Techniques

The dataset used for the evaluation of the proposed models is divided into three samples: a *Training set*, *Validation set* and *Testing set*. The training dataset is the one used to construct the model parameters by learning from the dataset by matching the input with the expected output. The validation set is mainly used to adjust the hyperparameters of the model and estimate the prediction error. Finally, the testing set is used to assess the performance of the system using the evaluation measures discussed earlier. There exist different cross-validation methods that can be utilized to validate the system performance using the dataset which are:

- *Holdout Method*:

  This approach is considered the simplest validation method where the dataset set is directly divided into three sets (training, validation, and testing). These sets are composed of a certain percentage of the data that are selected randomly. For example, if we choose 50% training, 10% validation and 40% testing, therefore, the model is built by using 50% of the dataset, the hy-

perparameters are adjusted with 10% of the dataset and the results of model performance is measured using the 40% of the dataset.

- *N-fold Cross-Validation (CV)*:

  This approach divides the data into N folds then it keeps repeating the process of training and validation by using one-fold of the data for validation and the rest of the data for training. This method allows all the data to get to be in a validation set exactly one time while in the training set (N-1) times. The advantage of using the N-Fold CV is that it helps to reduce both the underfitting and overfitting of the model as most of the data is used in both the training and validation set. Fig. 2.11 demonstrates the process of N-Fold CV with the calculation of accuracy. There are some special cases of the N-Fold CV that are commonly employed in the medical image analysis:

  - Leave-One-Out Cross-Validation (LOO-CV): This is a special case of the N-fold CV where N is equal to the number of samples in the dataset. Therefore, each sample (image or video in our dataset) is used once as a validation set alone and the rest as training. This validation is usually used when the number of samples in a dataset is considered small.

  - Leave-One-Patient-Out Cross-Validation (LOPO-CV): This type of validation is commonly used when the dataset provides details about samples gathered from each patient. In this case, the N fold are divided according to the number of patients in the dataset. Therefore, the image samples from one patient will never appear in the training and validation sets at the same time which leads to less bias results.

2.7 Summary

This chapter provides the clinical background for esophageal abnormality detection and the classification of endoscopic images/videos. The different types of abnormalities with the pathology stages are explained such as Esophgities, BE, EAC and SCC with illustrative figures. Moreover, the different endoscopic modalities that are used

for different purposes to examine the esophagus are described focusing on the two modalities (i.e. WLE, HD-WLE, and CLE) where our dataset is gathered from these tools.

The available datasets used in this thesis are represented. The CLE dataset used for the abnormality stage classification was obtained from *ISBI 2016* challenge dataset (*aidasub-clebarrett - Home* 2015). The HD-WLE image datasets used for the abnormality detection from images are gathered from the EndoVis textitMICCAI'15 challenge dataset (*Sub-Challenge Early Barrett's cancer detection* n.d.) and the open-access Kvasir (Pogorelov et al., 2017) dataset. The WLE video dataset used for the abnormality detection were acquired from the *GastroIntestinal Atlas* dataset (*El Salvador Gastrointestinal Atals* n.d.).

Furthermore, the evaluation protocols which will be used for the experiments are reported. For the assessment of the classification method, we adopt the standard evaluation metrics: *Senseititvty, Specificity, Precision* and *F-measure*. For the evaluation of the detection methods, we utilize the same measures as for the classification. Additionally, we employ the IoU (i.e. mainly evaluates generated bounding boxes in detection methods) as our main target from the detection methods is to locate the presence of abnormalities.

The next chapters will provide the details of the presented methodologies for abnormality classification and detection from the esophagus tube. Additionally, each chapter will explain and discuss the related work in the field of esophageal abnormality classification and detection methods.(i.e

Chapter 3

Esophageal Abnormality Grade Classification

3.1 Introduction

Automated digital pathology classification of a CLE image is considered to be a challenging process for several reasons. Although each stage has its histopathological characteristics, the transformation between each stage is considered visually small and difficult to identify easily. Moreover, the doctor examining the patients needs to be trained on the CLE imaging modality and is required to have background knowledge of histopathology. The motivation of this chapter is to develop a system that can automatically and accurately classify the different histopathological stages of the esophageal abnormality cell deformation in the esophagus tube focusing on IM (precancerous) and NPL (cancerous) stages as they are considered the important stages to diagnose early. The model also can serve as a second opinion for physicians and will help decreasing the number of biopsy samples needed for each patient.

A two-stage classification method presented in the literature by Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.) that classified the NPL in the first stage and then IM and GM in the second stage. Moreover, Veronese *et al.* (Veronese et al., 2013) suggested a method that extracts handcrafted features from patches within the images and uses all extracted features to classify the type of the CLE image. In this chapter, we proposed a unified framework that extracts selected handcrafted features directly from the full image and classifies the features in a single stage. The proposed model

first enhances the internal features of CLE images using an image enhancement filter that combines fractional integration with differentiation. Various features are then extracted on a multiscale level, to classify the mucosal tissue into one of its four types: normal squamous (NS), gastric metaplasia (GM), intestinal metaplasia (IM), and neoplasia (NPL). These sets of features are used to train two conventional classifiers: Support Vector Machine (SVM) and Random Forest (RF).

In this chapter, first we briefly describe supervised techniques including image features and classifiers. Secondly, we provide an overview of the state-of-the-art classification methods in the field of supervised-handcrafted based methods and deep learning-based methods. Followed by a detailed description of the proposed classification model. Afterward, we present the experimental setup, results, and discussion. Finally, this chapter is summarized.

The contribution of this chapter can be listed as follows:

- Enhancing the CLE images by improving the feature details using a combination of fractional integration and differentiation for a facilitated computerized classification and an improved visualization for physicians.
- Analysing the cell architecture and vessel properties of each stage to extract a powerful combination of selected handcrafted features for the classification process.
- Developing a novel unified framework that can automatically classify the captured CLE image in a real-time manner with a high accuracy result compared to the state-of-the-art methods.

3.2 Overview of Supervised Techniques for Proposed Methodology

Supervised methods are designed by extracting selected features from images/videos and building a model to find the relation between the extracted features and the

target. Manually/Visually selecting the set of extracted features from the image is named *handcrafted features*.

Generally, there are two main stages for any supervised model: the Training phase and the Testing phase. During the training phase; the model learns from the extracted handcrafted features to classify the image/region according to the ground-truth. Also, the internal parameters or weights are adjusted by the model. In the testing phase, unlabeled data are fed to the trained model to test the efficiency of the designed model by detecting/classifying the abnormal area. The supervised handcrafted feature models are common approaches in the literature for classifying and detecting esophageal abnormalities.

3.2.1 Introduction to Image Features

Handcrafted features are the set of features hand-picked by the data scientist. The process of selecting the appropriate features should be chosen according to the characteristics of each application. The image features are generally categorized into three types: color, texture, and shape.

- **Color Features**:

  The distribution of color within an image usually represents the color features within an image and is regularly visualized using the color histogram. The ability of the color features to characterize perceptual similarity colors is greatly influenced by the selection of the color space and color quantization scheme (W.-T. Chen, W.-C. Liu and M.-S. Chen, 2010). These features can be extracted from the different color spaces of an image. The target of the color space is to facilitate the specifications of the colors with a tridimensional coordinated system.

  In the literature, there exists different types of color space for color image processing such as RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), L*a*b, HSI (Hue, Saturation, Intensity), CMY (Cyan, Magenta, Yellow), YCbCr, YUV, etc.... More details about the different color spaces can be found in (Garcia-Lamont et al., 2018).

There are several descriptive statistical measures that represent the color features and extracted from the color space image, for example, mean, Variance (Var), Standard Deviation (STD), skewness, and kurtosis.

Furthermore, Color Coherence Vector (CCV) is used to avoid any similarity caused by the color histogram as it does not consider the spatial location of the pixel. CCV partition each bin in the histogram into two types; coherent and incoherent. Pixels are considered coherent if they are a part of a big uniformly-colored region and incoherent if they are part of a small uniformly-colored region.

- **Texture Features**: Texture refers to visual patterns or spatial arrangement of pixels that regional intensity or color alone cannot sufficiently describe. The texture is one of the important characteristics used in identifying objects or regions of interest in an image. It contains important information about the structural arrangement of surfaces. Because texture has so many different dimensions, there is no single texture representation method that is adequate for a variety of textures (Humeau-Heurtier, 2019). The texture descriptor methods can be classified into different categories; *statistical-based*, *structural-based*, *model-based* and *transform-based*.

  - Statistical-based: The statistical-based methods investigate the grey-level spatial relationship of textures and then extract some statistical features as texture description. In the literature, there exists famous statistical methods that compute texture according to the spatial organization such as *Grey level co-occurrence matrix (GLCM) (Haralick, 1979), Local Binary Pattern (LBP) (Ojala, Pietikäinen and Mäenpää, 2000), Grey level run-length matrix (GLRLM), Tamura features, Local energy patterns and Histogram of gradient magnitude*.

  - Structural-based: The structural methods break down textures into components such as texels that characterize the texture according to its spatial arrangement. There are two analysis methods for structural approaches, *bottom-up methods* in which texture primitives are decided then the spatial

arrangement is selected and *top-down methods* in which spatial structure is computed first then element extraction is presented.

- Model-based: The model-based approach uses mathematical models to represent texture. *Markov random field* (Cross and Jain, 1983) and *Fractal models* (Kaplan, 1999) are widely used for model-based texture representation.

- Transform-based: The transform-based extracts texture feature from an image by first representing the image in a frequency space or scale space, then extracts the content of the frequency and spatial domain. There are well-known transformation methods in the literature such as *Wavelet transform, Gabor transforms* and *Fourier transform.*

- **Shape Features**:

  Shape features is a very powerful feature that is related to the structure of details in the images. Shape features can be divided into two categories: *Countor based methods* that relies on the shape boundary points and *Region based methods* that uses shape interior points. The shape features can be defined as the center of gravity, mass, ratio, angle, and the number of edges (Manjula and M. Ahmed, 2017). Efficient shape features must have some important properties (Yang, Kpalma and Ronsin, 2008):

  - *Identifiability*: Shapes that have a similar appearance should have similar features that are different from other shapes.

  - *Translation, Rotation, and Scale invariance*: The extracted features must not be affected by the change of the shape location, rotation or scaling.

  - *Statistically Independent*: The compactness of the descriptors is confirmed if the features are statistically independent.

  - *Affine invariance*: The feature extracted needs to be invariant as much as possible with the affine transforms such as applying several translations, scales, flips, rotations, and shears.

- *Reliability*: The extracted feature must remain the same as long as the shape has the same pattern.
- *Noise Resistance*: Extracted features need to be robust against any kind of noise applied to the image.
- *Occultation invariance*: Extracted features must maintain their properties compared to original shape if parts of it are covered by other objects.

Edges are one of the shape descriptors related to the structure of details in the images mainly highlighted by boundaries (Patel and Tandel, 2016). Edge pixels are defined as locations in an image where there is a significant variation in gray level pixels in a fixed direction across a few pixels. They are one of the most important visual evidence for understanding images details (Kovesi et al., 1999). There are various methods found in the literature for edge detection methods such as Canny, Sobel, Prewitt, and Laplacian edge detector.

3.2.2 Supervised Classifiers

In this section, we will briefly explain the conventional classifiers that have been widely used in the literature and in this thesis for esophageal abnormality detection and classification.

- **Support Vector Machine (SVM)**:

  SVM is a learning algorithm that is originally introduced by (Cortes and Vapnik, 1995) and successfully extended by some researchers. SVMs are robust classifiers that have been widely used in different classification approaches. Several hyperplanes can separate between the data however the SVM classifier searches for hyperplane that produces the maximum distance (i.e. margin) to separate between classes. Hyperplanes are decision boundaries that support the data points classification. The larger the margin the more the data points are classified with confidence. A classification is represented by:

$$f(x) = w'x + b \tag{3.1}$$

where, w is the weight vector and b is the bias. For linear classification $f(x) = 0$. The hyperplanes obtained for the classes $y_i \in \{-1, 1\}$ thereby:

$$f(x_i) \begin{cases} \geq 0 \quad if \quad y_i = 1 \\ < 0 \quad if \quad y_i = -1 \end{cases} \tag{3.2}$$

The target of the SVM is to determine the hyperplane with a maximum margin with the minimum error known as *optimization*. The optimization for the linear separable classes is formulated as:

$$\min_{w,b} \frac{1}{2}||w||^2 \tag{3.3}$$

$$subject \quad to \quad y_i(w'x_i + b) \geq 1 \quad for \quad i = 1, 2, ..N \tag{3.4}$$

In the case of non-linear separation, the SVM requires a slack variable (ξ_i) to solve the optimization problem with error penalty (δ), specified by:

$$\min_{w,b,\xi} ||w|| + \delta \sum_{i}^{N} \xi \tag{3.5}$$

$$subject \quad to \quad y_i(w'\phi x_i + b) \geq 1 - \xi_i, \quad \& \quad \xi_i \geq for \quad i = 1, 2, ..N \tag{3.6}$$

According to the application, a kernel method is used by the SVM such as *linear, polynomial kernel, Radial Basis Function (RBF), hyperbolic tangent, sigmoid*; are adopted to automatically realize a non-linear mapping of the feature space to maximize the margin hyper-plane. The chosen kernel is defined as *K(x,y)*.

- **Random forest (RF)**:

  RF is a classification method that deploys an ensemble decision trees that was first introduced in (Liaw, Wiener et al., 2002). It is composed of a selection

of tree classifiers where each classifier randomly selects a subset of the input vector and each tree votes for the highest selected class to categorize the input. There are two types of randomness developed within the trees. First, random samples from the input data are used to build a tree. Secondly, a subset of the features is randomly picked to create the best split at the tree node. During training, different parameters are initialized in a RF classifier such as:

- *Depth of Tree (D_tree)*: It represents the maximum number of nodes determined from the root to any leaf of the tree.
- *Random seed point (rp)*: The value of *rp* is responsible for controlling the amount of randomness utilized during the training of the trees.
- *Forest size*: The number of trees in the ensemble.
- *In bag Fraction (f)*: To train a tree, a fraction f from the total training set is used.

As described, the RF requires essential parameters that need to be initialized. If these values are set empirically it can lead to leakage of the data. However, RF classifiers have different advantages; it presents a strong prediction performance and is less prone to overfitting. Also, it has a fast classification run time as trees can run in parallel.

3.3 Overview of the Classification Methods Available in the Literature

There exists a few amount of research in the literature for automatic esophageal grade classification from CLE endoscopic images. In this section, we will be reviewing the methods that utilize handcrafted features with conventional classifiers for the classification process.

A patch-based classification method was suggested by Grisan *et al.* (Grisan, Veronese et al., 2012) to distinguish between the IM and GM regions in the same CLE image. The method first extracted rotation invariant local binary patterns (RLBP)

and contrast features from each patch. Then, patches with contrast value below a certain threshold were eliminated from the analysis and labeled as "ungradable". The features from the remaining patches were used to train a support vector machine (SVM) based on LOO-CV. The result showed 98.85% sensitivity and 65.22% specificity for detecting the IM class, which was considered efficient in classifying specific regions inside the image.

Later on, Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.), introduced a computer diagnosis method for classification between IM, GM, and NPL CLE images that achieved an overall accuracy of 82%. In this method, the features are extracted from an image-based approach and processed on a two-stage classification. In the first stage, images were classified as either NPL or not. Images from the non-NPL class, from the first stage, are passed to the next stage where they are classified as either IM or GMP, based on a proposed leakage pattern extraction. The evaluation of the classification performance was based on the LOO-CV on 336 CLE images.

Veronese *et al.* (Veronese et al., 2013) employed a hybrid patch-based and image-wide classification approach to classify CLE images on two stages into IM, GM, and NPL grades. In the first stage, a patch-based classifier is used to extract intensity distribution values, geometric characteristics, and rotation invariant LBP to classify whether the image is IM or not. In this stage, a voting scheme is used the classification task. If the number of positive sub-blocks in one image is higher than a certain threshold, then the images are categorized as IM. In the second stage, a studied leakage pattern method extracts different features from the non-IM image (i.e. known from the first stage) to classify them into GM and NPL grades. This model achieved an overall accuracy of 96%.

Recently, Nadri *et al.* (Nardi et al., 2019) proposed a classification method for BE surveillance. As a preprocessing phase, the concept of the Local density function (LDF) is employed by the model to determine cellular structures in case of illumination changes. Followed by that, a combined set of texture features are extracted to classify between NS, GM and IM categorize. The texture features are composed of fractal features extracted from different level sets of the LDF and LBP features

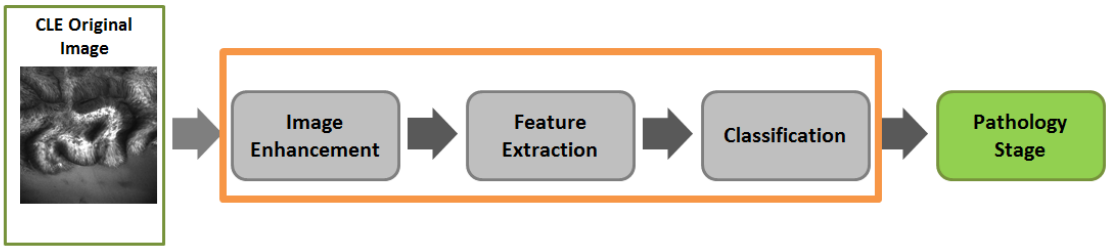


Figure 3.1: The framework of the proposed classification model. The input image is first enhanced through the proposed filter. Then different features are extracted to classify the pathology class of the image.

extracted at different radius orientations. Using the LOPO-CV, the suggested model achieved an accuracy of 88.5% using the SVM and 89.2% when using the RF classifiers.

Only one deep learning method to identify IM, GM and NPL was put forward by Hong *et al.* (Hong et al., 2017), designing a convolutional neural network (CNN) composed of 4 CNN layers with two max-pooling layers in between and two fully connected networks at the end. The overall accuracy of the system based on the testing images was only 877%. The model was trained on a limited size of the dataset even after applying augmentation. The dataset before augmentation is composed of 235 classified as 155 IM, 26 GM, and 55 NPL. The small amount of GM images trained led to the failure of the model in classifying any of the GM images during testing. Additionally, the results of testing the network are based on a very small imbalanced data sample that consists of 26 images only (22 IM, 0 GM, and 4 NPL). The accuracy results don't imply the efficiency of the proposed network as it was tested on a small sample of the dataset without the GM class.

3.4 Methodology

This section will explain the details of the proposed method for the CLE image classification. The pipeline of the proposed classification model consists of three steps as illustrated in Fig. 3.1. First, the CLE image is enhanced using a proposed novel filter, then handcrafted features are selected and extracted from each image.

Finally; images are classified into its pathology stage. Each of these steps will be described in detail in the following sections.

3.4.1 Overview of the Framework

Fig. 3.2 represents the overall framework of the proposed classification method. A novel enhancement filter is first applied to the input image to improve the internal features of the image. Then a different set of features are extracted on a multistage level to discriminate between the four stages. The handcrafted features are selected according to the properties of each stage to facilitate the differentiation between each grade. Afterward, the extracted features are classified by using the *SVM and Random Forest* (i.e. separately) to classify the pathology grade. Each of these steps will be described in detail.

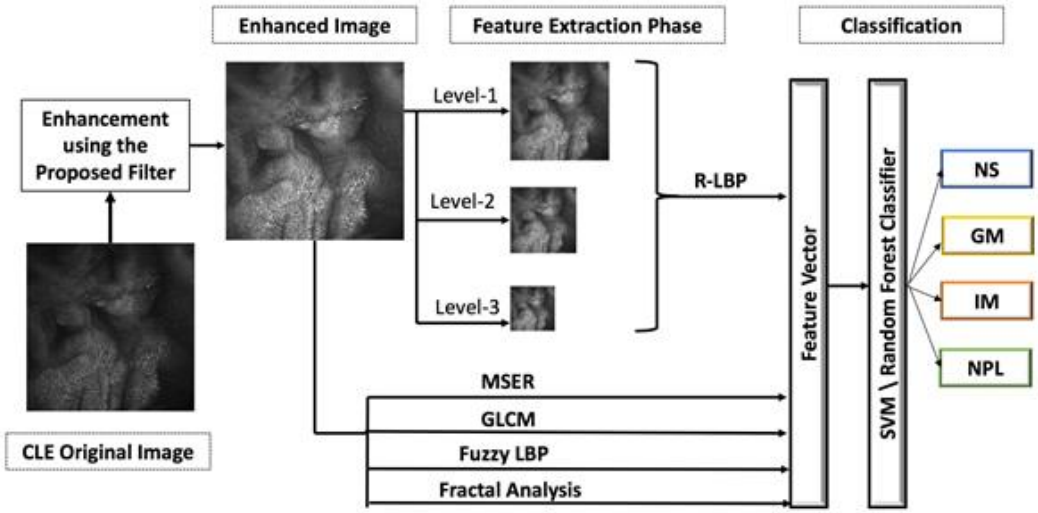


Figure 3.2: The detailed proposed classification method. A post-processing enhancement filter is applied to the input image. Then multiscale features are extracted from each enhanced image. Finally, images are classified into the grade deformation.

3.4.2 Enhancement Phase

In the first phase of the proposed model, the CLE image is enhanced by applying a novel digital filter that utilizes the *Fractional Differential* (**FD**) and *Fractional Integration* (**FI**) in the wavelet sub-bands. As shown in Fig. 3.3, the proposed filter

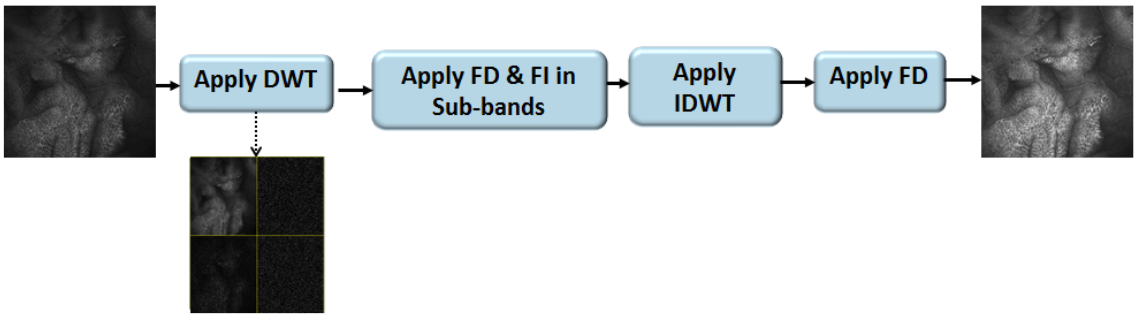


Figure 3.3: Proposed enhancement filter to improve the features of the input image as a preprocessing phase.

firstly decomposes the image into its Discrete Wavelet Transform (DWT), dividing it into four sub-bands (LL, LH, HL, and HH). Then, FI is applied to the diagonal sub-bands (LH-HL) to remove the noise, while the FD is applied to the HH sub-band to improve selected texture features. The improved image is reconstructed by applying the Inverse DWT (IDWT), and then the FD filter is re-applied on the whole reconstructed image to improve the overall texture. In the following subsections, each phase will be explained in detail.

DWT

The DWT is a special case of the Wavelet Transform (WT) that provides a compact representation of a signal in time and frequency through two filters:

- A high-pass filter where high-frequency information is saved, low-frequency information is lost.
- A low pass filter where low-frequency information is saved, high-frequency information is lost.

The DWT decomposes the image into four different frequency sub-bands holding the majority of the data position and emphasizing the features. These sub-bands correspond to approximate, horizontal, vertical, and diagonal features, respectively. As shown in Fig. 3.4 the sub-bands are named as LL, LH, HL & HH (i.e. L=Low, H=High). The LL sub-band is approximately located at half the original image, while the HH sub-band contains the high-frequency details of the image. On the other hand, the HL-LH holds the changes to an image. For the one level decomposition,

the 2D DWT of the image function $f(x, y)$ is written as (Wu and L.-G. Chen, 2001):

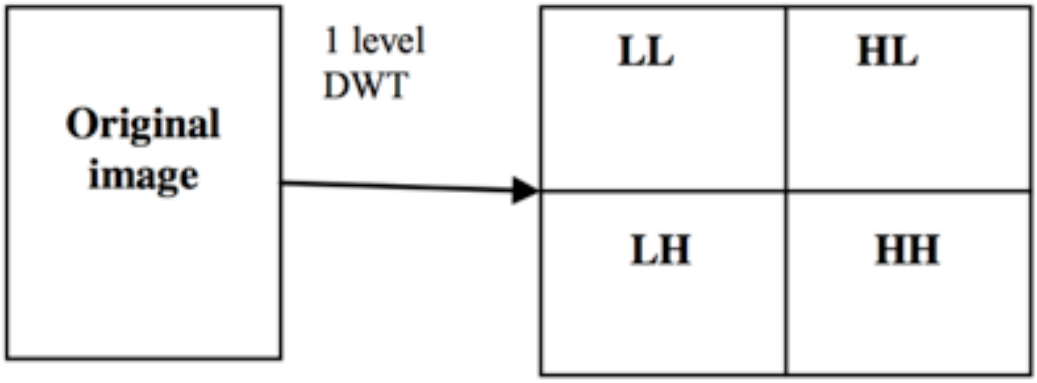


Figure 3.4: Illustration of DWT 1-Level Transform

$$W_\phi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)\phi_{j_0,m,n}(m, n) \tag{3.7}$$

$$W_\psi(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)\psi^i_{j_0,m,n}(m, n), \quad i = \{H, V, D\} \tag{3.8}$$

where,*phi*irepresnets the scaling function, $\psi(t)$ is a time function with finite energy and fast decay called the mother wavelet. The DWT is generally used to improve the features (Youssef, ElFarag and N. M. Ghatwary, 2014). It allows selective and separate suppression of coefficients in the different sub-bands, thus affects low-frequency, high frequency, and directional features differently. We empirically chose Daubechies (db2) (Daubechies, 1992) as the mother wavelet of the DWT analysis at level 1 decomposition. More information about DWT can be found in (Kingsbury, 1999). Fig. 3.5 illustrates sample from the CLE images from our dataset after applying DWT.

Fractional Differential (FD) and Fractional Integration (FI)

FD and FI are mathematical operations related to the field of fractional calculus that deals with non-integer values (Almeida, Tavares and Torres, 2019). FD has proven in the literature to provide better performance in improving the texture of images than other methods (Pu, Zhou and Yuan, 2010) while FI has shown to be an

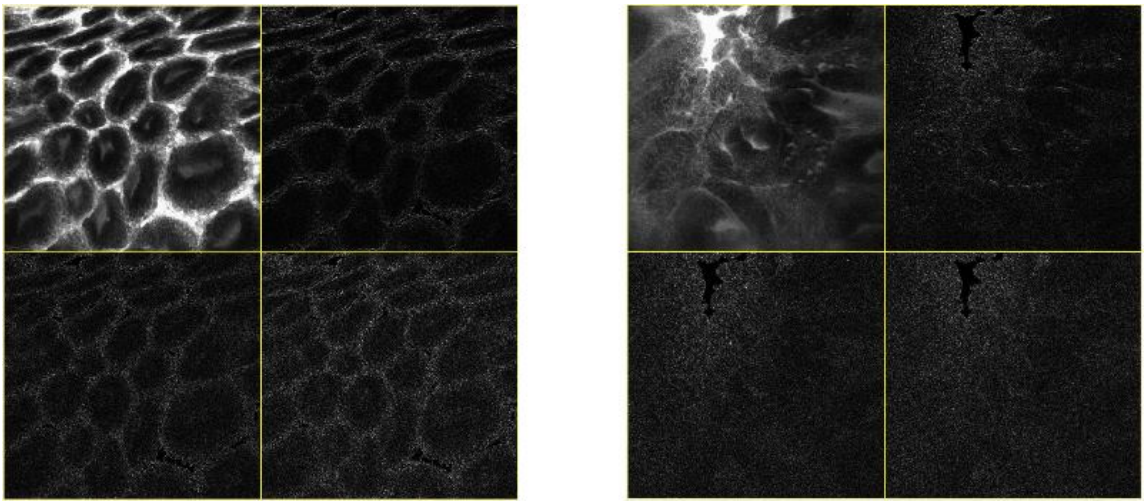


Figure 3.5: Example of DWT 1-Level Transform for CLE images.

effective image noise removal method maintaining image feature details (Jalab and Ibrahim, 2012).

In medical image processing, the texture is one of the key features that can improve the performance of classification. Texture can help in describing the positioning and local spatial variation of pixel intensity (Castellano et al., 2004). Applying integer-order differentiation arising from the discrete nature of the image may result in the disturbance of the fine textural details that we need to capture. Therefore, using the FD is an efficient method to deal with the texture like problems. In our model we apply the FD twice: first, it is applied to the HH sub-band to improve the high-frequency details of the image. Secondly, it is applied to the overall image after reconstruction from the DWT to enhance the overall texture details of the image. The FD is implemented using a mask filter based on equation 3.9 inspired by (Q. Yu et al., 2013):

$$\frac{ds(x,y)}{dx} = -\frac{1}{(2\cos(2\Pi\alpha)h^{\alpha})}\sum_{d=0}^{n} M_d s(x-dh,y) \qquad 0<\alpha<1 \tag{3.9}$$

$$M_0 = -\frac{\Gamma(1-\alpha/2)}{\alpha\Gamma(1-\alpha/2)\Gamma(-\alpha)} \tag{3.10}$$

$$M_d = \frac{(-1)^{d+1}\Gamma(\alpha/2)\Gamma(1-\alpha/2)}{\Gamma(\alpha/2-d+1)\Gamma(\alpha/2+d+1)\Gamma(-\alpha)} \quad d=\pm1,\pm2,.. \tag{3.11}$$

where, α is the derivative order of the fractional differentiation that takes a non-integer value ranging for $0<\alpha<1$, M is the applied mask with a window size of $[n*n]$ and is calculated based on equation 3.10 and 3.11, with d is the direction where the masked is applied.

On the other hand, Denoising is important to remove the noise from the image while preserving the quality of its features. So, the FI is applied at the LH-HL sub-bands where they hold the changes of images or edges along with vertical and horizontal directions, respectively. The FI can remove noise and sustain the texture and edge features in an image (Guo et al., 2012). In the proposed model we utilize the FI mask suggested by (Jalab and Ibrahim, 2013) for the enhancement phase.

3.4.3 Feature Extraction

Features are calculated based on the properties of the histopathology stage. Each grade NS, GM, IM, and NPL has a particular internal structure (i.e. as described earlier in Chapter 2 Sec. 2.3.5). Moreover, the computation complexity is also taken into consideration as the CLE is an in-vivo technology; so the automatic classification process needs to be performed in a real-time manner. An aggregation of texture and intensity features are calculated as below:

Gray Level Co-occurrence Matrices (GLCM):

GLCM is a statistical method that examines the texture in an image by examining the spatial relationship between pixels. It captures the second-order statistical features for texture. GLCM uses the co-occurrence matrix to statistically characterize the way certain grey-level pairs occur in relation to other grey-levels in a specified spatial relationship. They correlate the related frequencies (f) at location (x, y) with distance (d) and direction (θ). There exist two kinds of co-occurrence matrix; the first has asymmetric matric where each pair is separated by range from d to -d in the direction of θ. The second case counts only pairs separated by the distance d, therefore, the output is a square matrix that has the dimension of intensity values in the image. An example of the co-occurrence matrix from both types is shown in Fig. 3.6. A small d value is used to find fine texture details in an image while larger

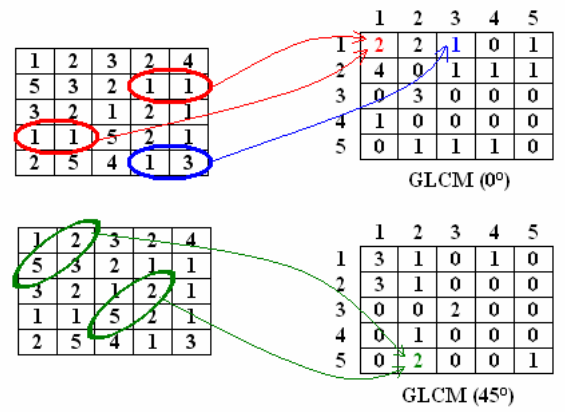


Figure 3.6: An example of generating a GLCM matrix using the two types. The above figure shows an example of finding similar pairs with spatial distance d=1 between pixel pairs. The lower figure illustrates an example of finding similar pairs with $\theta = 45$ and spatial distance d=1 between pixel pairs (Tou, Lau and Tay, 2007).

d is required to classify coarse texture details. An image with highly correlated pixel values produces a matrix with most pairs grouped along the diagonal. Fourteen textural features were defined by Haralic *et al.* (Haralick, 1979), computing different properties to obtain GLCM texture features.

One of the dysplasia properties is that it usually has a high entropy value. Moreover, low contrast and homogeneity of pixel pairs which helps to differentiate the degree of dysplasia. For that reason, the following GLCM features (*Entropy, Contrast, Homogeneity*) are utilized in our model:

$$Entropy = \sum_{i,j=0}^{N-1} -ln(P_{ij})P_{ij} \tag{3.12}$$

$$Contrast = \sum_{i,j=0}^{N-1} P_{ij}(i-j)^2 \tag{3.13}$$

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1+(i-j)^2} \tag{3.14}$$

where P_{ij} is the element of normalization between two pixels i and j, N is the number of grey levels in the image.

MP-RLBP

A multi-scale feature named **MP-RLBP** is proposed. It extracts the Rotation Local Binary Pattern (RLBP) from different levels of Gaussian pyramid images. *Gaussian Pyramid* is a multi-scale representation of an image that samples the image down into smaller groups of pixels. The Gaussian pyramid is constructed by repeatedly calculating the average weight of neighbored pixels by convolving the original image with the Gaussian function. As shown in Fig. 3.7, Gaussian Pyramid can be visualized by stacking smaller versions of the image on top of one another. This method produces a pyramid shape, where, the original image is the base of the pyramid and the tip is a single-pixel representing the average value of the entire image. The Gaussian pyramid of an input image (I) is defined as:

$$G_0(x, y) = I \tag{3.15}$$

$$G_{i+1}(x, y) = R(G_i(x, y)) \tag{3.16}$$

where, R is the *reduce* process of convolving the image with a Gaussian low pass filter. The design of the filter is set that the center pixel takes more weight than the neighboring ones while the sum of the remaining is set to 1. The gaussian pyramids are easy to compute, useful for multi-scale edge estimation and provide useful information in the finer scales for texture mapping.

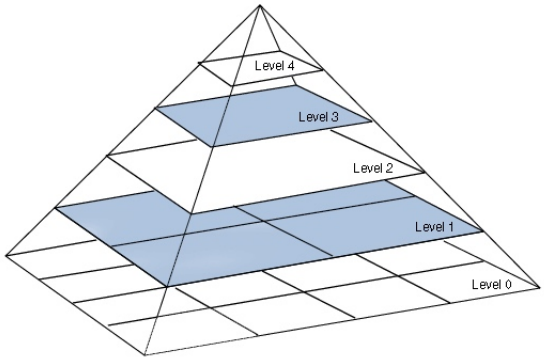


Figure 3.7: Example of Gaussian pyramid representation.

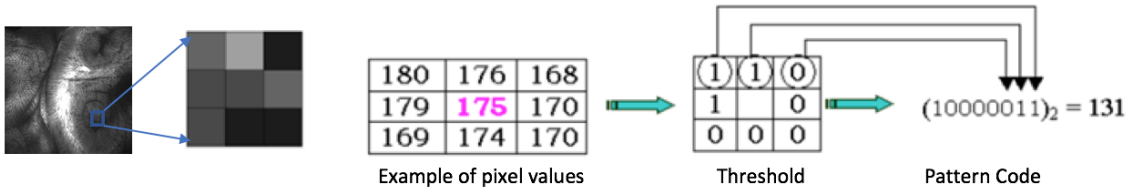


Figure 3.8: Example of LBP operation for a pixel with neighborhood (3×3)

The RLBP (Mehta and Egiazarian, 2013) is an extension of LBP that was first represented in (Ojala, Pietikäinen and Mäenpää, 2000). The LBP is an efficient texture descriptor for images that represent each image pixel (p_i) with a binary pattern according to neighboring pixel values. The calculated values are based on the difference between the grey value of the current pixel (p_i) and the neighborhood pixel values $(n \times n)$. It has two essential elements, P the corresponding pixel count and R the radius length centered around (p_i). The LBP codes are measured by:

$$LBP_{P,R}(p_i) = \sum_{p=0}^{P-1} s(y) \times 2^p \qquad s(y) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases} \tag{3.17}$$

where y represent the difference of the intensity levels between the neighboring pixels (p_p) and the neighborhood center pixel (p_i) (i.e. $y = p_p - p_i$). A binary value from 0 to 255 is gained by concatenating the values of the neighborhood results in a clockwise or anti-clockwise direction for each pixel. An example of LBP operation for a neighborhood (3×3) is illustrated in Fig. 3.8.

The RLBP takes into consideration the rotation changes where it is computed by shifting the output binary code circularly by setting a local reference direction (D) in every circular neighborhood and calculate the descriptor in reference to it. The value D is specified as the index of the neighborhood pixel that has the maximum difference value with centered pixel (p_i) which is set as the reference for the neighborhood weights. The D is defined as:

$$D = \underset{p \in (0,1,2,\ldots,P-1)}{\arg\max} |p_p - p_i| \tag{3.18}$$

The neighborhood rotation takes place concerning its center shift based on the direction D by the same angle (θ). Therefore, RLPB is defined as:

$$RLBP_{P,R}(p_i) = \sum_{p=0}^{P-1} s(y) \times 2^{mod(p-D,P)} \tag{3.19}$$

where, *mod* represents the modules operation and the weight term for $2^{mod(p-D,P)}$ is defined according to the value D. Moreover, the weights are circularly shifted with respect to the D direction which leads to rotation invariant.

To extract the MP-RLBP features, the image is first decomposed into N-levels using a Gaussian Pyramid (Adelson et al., 1984). The RLBP is extracted from each scaled image as shown in Fig. 3.9 to measure the relationship between a pixel and its neighbor as a descriptor. The N-level of the Gaussian pyramid in the proposed model is adjusted to level-3 while the parameters of RLPB were set to R=4 with P=8.

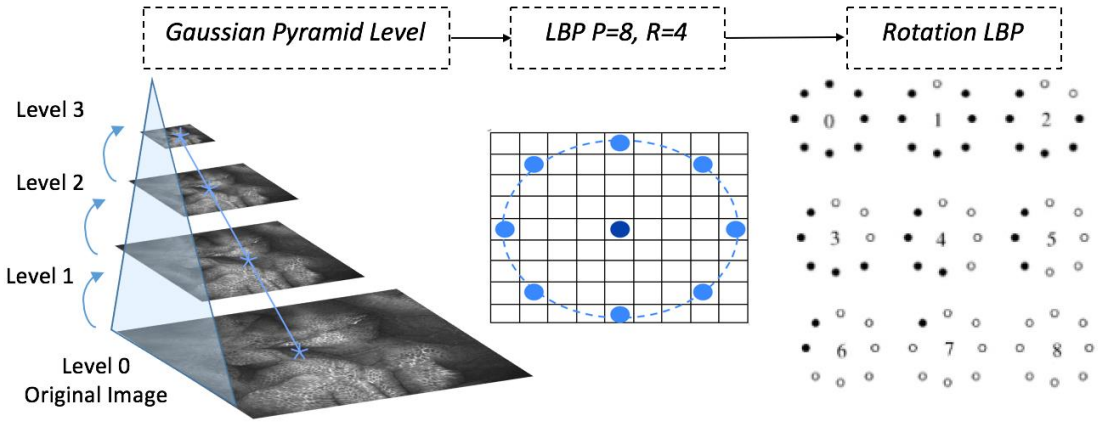


Figure 3.9: Example of MP-RLBP extraction from CLE Multi-Scale Pyramid Image.

MSER

MSER is known as a shape descriptor that was first introduced by Matas *et al.* (Matas et al., 2004). It can detect regions having different properties by evaluating the stability of extremal regions which represent the high and low-intensity regions compared to all pixels of the outer boundaries (Li and Yin, 2016). MSER is considered a fast region detector with a good performance for the homogeneous regions

with distinctive boundaries in an image. It has four main parameters: *(threshold (t), minimum (minR) and maximum (maxR) size of each region and maximum stability function q)*, defining q as:

$$q(R_t) = \frac{A(R_t)}{\frac{\partial}{\partial t}A(R_t)} \tag{3.20}$$

where, A expresses the area of the region R at threshold t. The image features detected by MSER are the stable regions that are mapped into a global high-dimensional feature vector of size 64. As previously explained, each BE grade type has a certain deformation of the cell properties based on the stage. Therefore, extracting MSER features help improve the accuracy of classification. In the proposed model, we empirically set the variables of the MSER to $t = 2$, $minR = 30$, $maxR = 1400$ and $q < 25$.

Fractal Texture Features

The calculated set of features includes the ***Fractal texture features*** as presented by Costa *et al.* (Costa, Humpire-Mamani and Traina, 2012). This feature measures the fractal dimension using the box-counting method, mean grey level and size (pixel count) from a set of binary images. The binary images are created using a two-threshold decomposition that characterizes the texture patterns of the CLE input image. Image boundaries are then extracted from each binary channel using edge detection. Finally, fractal features are computed using the binary edge channels. The extracted features are Area, Intensity and Fractal Dimension. The *area* that represent the number of edge pixels available in the CLE image. The *intensity* calculates the mean intensity of the CLE image corresponding to the edge pixels. Finally, the fractal dimension measures the structure complexity if an image and is calculated from the boundary image using the following equation:

$$D_0 = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log \varepsilon^{-1}} \tag{3.21}$$

where $N(\varepsilon)$ expresses the counting of hyper-cubes (rectangles in the case of 2D space)

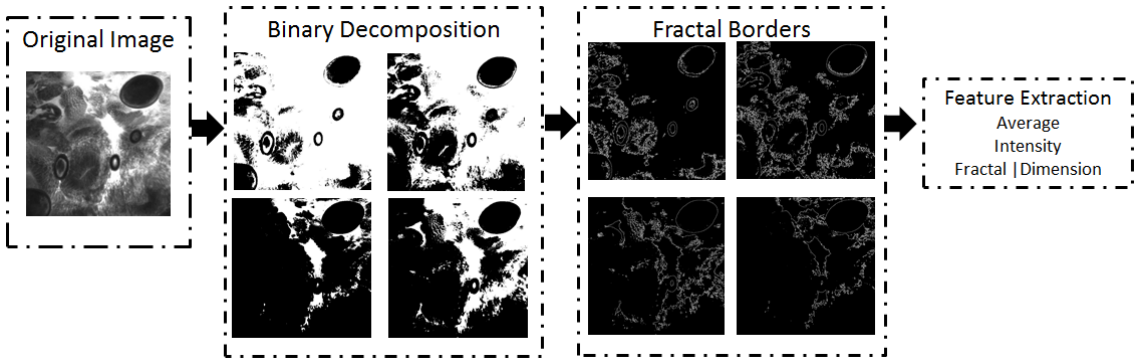


Figure 3.10: The process of extracting fractal texture features from CLE image.

of dimension E and length ε. Fig. 3.10 represents the process of extracting fractal texture features from the CLE image.

Additionally, we compute the ***Lacunarity*** which measures the spatial distribution of the fractal gaps (i.e. related to the size distribution of the holes). Lacunarity is an equivalent measure to the fractal dimension that describes the texture of a fractal. The Lacunarity is not related to the topology of the fractal and needs more variables to be fully defined. The low lacunarity represents a homogenous texture where all gaps represent the same size, on the other hand, high lacunarity provide heterogeneous texture. Together, the lacunarity and fractal dimension define patterns extracted from images. Our model will benefit from this feature to identify the gaps caused by the vessels appearance and compare high complex details in similar stages (i.e. such as GM and IM).

FLBP

FLBP is another extension LBP. The LBP measures the relationship between pixel intensity and its neighboring intensities. The fuzzy logic deals with the uncertainty of the LBP and improves the textures classification by employing a set of fuzzy rules (Youssef, ElFarag and N. M. Ghatwary, 2014). The FLBP is described in (Iakovidis, Keramidas and Maroulis, 2008), where two membership functions were implemented according to two fuzzy rules to extract the texture descriptor. The two rules presented to express the relation of intensity values of neighborhood p_i and the central pixel p_{center} with certainty degree d_i defined as:

Rule (R_0): If $p_i < p_{center}$ therefore $d_i = 0$

Rule (R_1): If $p_i > p_{center}$ therefore $d_i = 1$

Accordingly, two membership functions m_0 and m_1 are required based on the above two rules (R_0 and R_1). The functions m_0 and m_1 are responsible to define the degree d_i to 0 and 1 respectively. As shown in Fig. 3.11, let function m_0 define the degree $d_i = 0$ when the value of p_i has a smaller value thanp_{center} and m_1 define the degree $d_i = 1$ when the value of p_i has a greater value thanp_{center}. There exists a parameter $T \in [0, 255]$ that controls the degree of fuzziness for both m_0 and m_1. The membership function are defined as:

$$m_0(i) \begin{cases} 0 & if \quad p_i \geq p_{center} + T \\ \frac{T - p_i + p_{center}}{2 \times T} & if \quad p_{center} + T > p_i > p_{center} - T \\ 1 & if \quad p_i \leq p_{center} - T \end{cases} \tag{3.22}$$

$$m_1(i) = 1 - m_0(i) \tag{3.23}$$

For a neighborhood of size ($n \times n$), the contribution C_{lpb} of the LBP code in a single bin of the FLBP histogram defined as:

$$C_{LBP} = \prod_{i=0}^{2^n} m_{d_i}(i) \tag{3.24}$$

where, $d_i \in \{0,1\}$ and the LBP (eq. 3.17). For each area around pixel i, the value of d_i can be 0 or 1 with a grade of m_0 and m_1 that results in different contributions of eq.3.24. Therefore, each neighborhood provides more than one bin in the FLBP histogram representing a total number as:

$$\sum_{LBP=0}^{255} C_{LBP} = 1 \tag{3.25}$$

The main advantage of FLBP features that they can differentiate between strong and weak patterns. Since the cell texture representation is very challenging and in

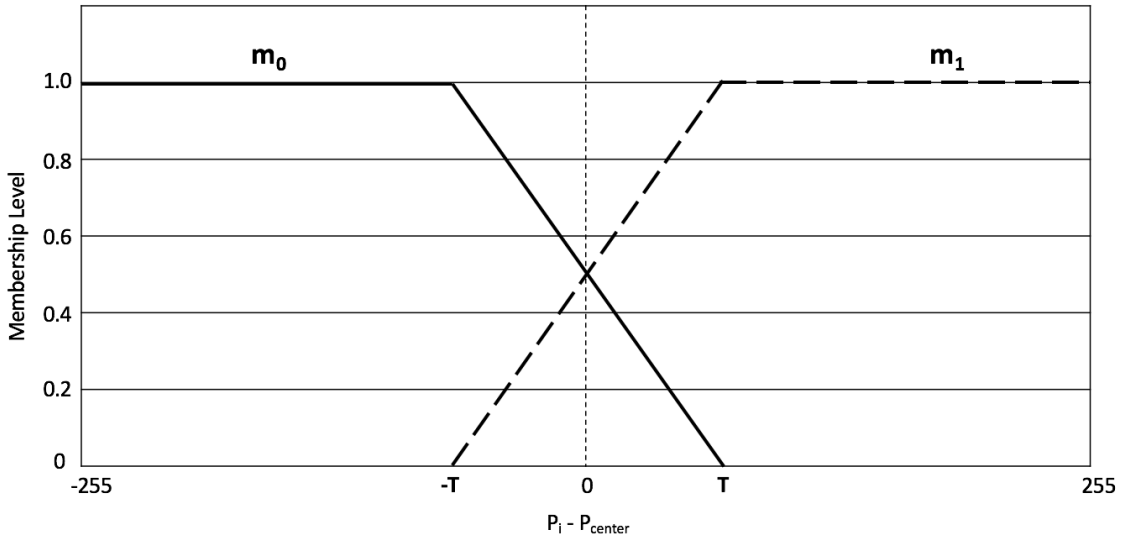


Figure 3.11: Membership functions m_0 and m_1 as a function of $p_i - p_{center}$ for T values.

each consecutive stage there exists a very high similarity in their properties, a feature such as FLBP will then have the ability to measure texture information.

3.4.4 Classifiers

For the classification, we employ the SVM and RF classifiers as they proved to have a good performance in similar applications in the literature (i.e. as explained in previously in Sec. 3.2.2). As described, two important hyperparameters are employed for the *SVM* classifier: the cost parameter C and the kernel function $K(x_i, x_j)$. In our model we evaluate two commonly used kernel functions; the polynomial kernel and Radial Basis Function (RBF) expressed as:

- Polynomial kernel:

$$K(x_i, x_j) = (x_i * x_j + c)^d \quad d > 0 \tag{3.26}$$

- RBF

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2) \quad \gamma > 0 \tag{3.27}$$

where, c is a parameter that trades off between the impact of the high-order parameter against the lower-order ones, and d represents the degree of the polynomial

that relates the sum of the supported variables. After evaluation, the polynomial kernel showed better performance where the results will be presented in the next section. The variables of the SVM polynomial kernel have been set to c=1 and d=2. The main parameters for the RF classifier are: the tree depth (D_tree), random seed point (rp) and bag Fraction (f). It is composed of a selection of tree classifiers, where each classifier randomly selects a subset of the input vector and each tree votes for the highest selected class to categorize the input. In the experiments, the parameters of the RF classifier were set to D_tree=100 and rp=1.

3.5 Experimental Setup and Results

In this section, the evaluation measures for the proposed model and enhancement filter are presented. Afterward, the dataset used, implementation details and evaluation protocols are described. Finally, comprehensive experimental results are presented and discussed in terms of quantitative and qualitative evaluations.

3.5.1 Evaluation Measures

Classification Evaluation Measures:

To evaluate the performance of the proposed model in classifying esophageal abnormality cell deformation stages we employ the standard performance metrics generally adopted in medical image classification *Accuracy, Sensitivity, Specificity, Precision* and *F-Measure* which was explained in details in Chapter 2 (Section 2.6, Equations 2.1 to 2.5).

The LOPO-CV is used to train and test the model for all the experiments and to compare it with the state-of-the-art models.

Enhancement Filter Evaluation Measures:

To assess the performance of the enhancement filter objectively, we utilize two well-known image quality quantitative measures: *Contrast Improvement Index (CII)* and *Tenengrad Measure.*

- Contrast Improvement Index (CII): measures the improvement of the contrast between the enhanced and original image.

$$CII = \frac{A_E}{A_I} \tag{3.28}$$

where, A_E is the average values of local contrast C from the enhanced image and A_I from the original image. The local contrast C is calculated from a window size of 3×3 as: $\frac{maximum-minimum}{maximum+minimum}$. The increase of the CII values indicates an improvement in the contrast of the enhanced image.

- Tenengrad Measure: is used to examine whether structural information in the enhanced image has been improved or not, therefore, it is one of the most accurate and robust measures for image quality evaluation. For each enhanced image E, the gradient $\triangle E(i,j)$ at each pixel location (i,j) is used to calculate the Tenengrad value where the partial derivatives are acquired through a high-pass filter using Sobel operator, with the convolution kernels e_i and e_j. The gradient magnitude is defined as:

$$S(i,j) = \sqrt{(e_i \times E(i,j))^2 + (e_j \times E(i,j))^2} \tag{3.29}$$

and the Tenengrad value (T) for an image is calculated as:

$$T = \sum_i \sum_j S(i,j)^2 \qquad S(i,j)^2 > t \tag{3.30}$$

where t is a threshold. A larger Tenengrad value implies a higher quality of an image.

3.5.2 Dataset and Implementation

The model is evaluated using the CLE dataset explained in Chapter 2 (Section 2.5.1). The dataset consists of 557 images gathered from 96 patients with four histopathology stages: NS, GM, IM, and NPL.

For the implementation and experimental evaluation, the Matlab_R2016a has been used on a 2.9 GHz dual-core Intel Core i5 with 8.0 GB SDRAM. In our study, we evaluated and tested different α values (i.e. ranges from $0<\alpha<1$) for the enhancement filter in several directions with different window sizes. The best performance was found to be with the window size of the enhancement filter mask (M) is adjusted to 5×5 and applied in d=8 directions. Additionally, the FD in the HH sub-band α is set to 6 while when applied to the whole image it is set to α=4.

Furthermore, the extracted features were concatenated to form a feature vector of size *(620×1)* made up of: *(MP-RLBP(286×1), MSER(64×1), GLCM(3×1), Fractal Texture Features(11×1), FLBP(256×1))* to be used for classification.

3.5.3 Experimental Results and Discussion

To support the doctors with a valid second opinion, the proposed model is concerned with the accuracy of automatically classifying each histopathology grade, specifically the precancerous stage IM and later NPL stage. Several experiments were conducted in this section using the CLE dataset. In the first experiment, we evaluate the performance of the proposed model using the dataset when classified with two different classifiers (*SVM & RF*). Secondly, we assess the performance of the novel enhancement filter by comparing it with different well-known standard filters. Moreover, we compare the performance of the model with state-of-the-art methods.

The confusion matrix in Table 3.1 illustrates the performance of the proposed model using the LOPO-CV with the SVM classifier that showed a better performance than the Random Forest as will be discussed. Since each patient might have more than one image, this type of validation is more efficient to measure the confidence of the model and avoid any bias classification. Based on this validation method, the model was able to achieve an overall accuracy of 96.05% with a sensitivity of 97% for IM, 90% for GM, 94% for NPL and 100% for NS. The results show that the misclassified images are mostly classified incorrectly as a higher grade. Therefore, the system is considered better than misclassifying any true positives that need to be examined.

Moreover, experiments have been applied to the dataset without using the filter to

Table 3.1: Proposed Model Confusion Matrix using LOPO-CV on the 96 patients with SVM classifier

	IM	**GM**	**NPL**	**NS**	Sensitivity (%)	F-Measure (%)
IM	389	1	12	0	97.0	97.0
GM	3	37	1	0	90.0	92.0
NPL	3	1	64	0	94.0	88.0
NS	0	0	0	45	100.0	100.0
Specificity	96.0	99.0	97.0	100.0	Accuracy = 96.05%	
Precision	98.0	94.0	83.0	100.0		

evaluate the efficiency of the proposed enhancement filter on the classification results, we extracted the suggested features from the original image (without enhancement) and classified using both the SVM and Random Forest to evaluate the effect of the filter. The results of each classifier are compared together in Table 3.2, and the sensitivity for each class and overall accuracy is illustrated in Fig. 3.12. Starting with the **IM** class, the SVM classifier was able to detect more IM images accurately with less false positives when compared to the RF (with or without filter) showing the highest values throughout the table for all the evaluation measures. Moreover, applying the enhancement filter to the images increases the sensitivity from 95% to 97%, specificity from to 79% to 96%, precision from 94% to 96% and F-measure from 96% to 97% when using the SVM classifier. While in the case of using the RF, results using enhancement increased the sensitivity from 86% to 90%, specificity from to 88% to 91%, precision from 95% to 96% and F-measure from 91% to 94%.

Followed by the GM class, using the RF classifier in enhanced images outperformed with a result of 100% while a better result for the specificity was shown when using the SVM classifier, indicating that SVM was able to decrease the number of false positives for this class. As shown in the table, using the filter for both classifiers showed a significant increase in the results of all evaluation measures.

Pursuing with the results of **NPL** class, the sensitivity and F-measure with values of 94% and 88% using the SVM on enhanced images surpassed the results from RF (with and without filter) and SVM without the filter. The specificity for the RF

classifier on the enhanced images was better than the other three values. On the other hand, the precision value 88% of the SVM without the filter was the best in this case. This was the only incidence where the experimental results for non-enhanced images showed a better performance than enhanced images throughout the table.

Finally, the **NS** class results were improved when using the filer for both classifiers resulting in an accuracy of 100% without allowing any other classes to be misclassified as NS. As a conclusion from this comparison, the SVM classifier on enhanced images was more efficient compared to RF for classifying the four pathology stages. Therefore, it will be used for the rest of the evaluations made in this section.

Furthermore, to evaluate the performance of the proposed enhancement filter on improving the quality of the image, we employ different quantitative measures, as discussed in the previous section. Table 3.3 illustrates the performance measure values obtained after applying the proposed filter in comparison with different standard enhancement techniques: *Histogram Equalization (HE), Adaptive HE, Median Filter, Wiener Filter* and *Gaussian Filter*. From the table, it can be seen that the proposed enhancement filter gives a higher CII value compared to the standard enhancement methods showing that the filter can provide better contrast within the image. Additionally, throughout the table, the Tenegard value outperforms against the other conventional filters, therefore, we can conclude that the structural information has been improved which leads to an improved classification result. In addition to the quantitative evaluation results, we also demonstrate some qualitative results in Fig. 3.13 that represent an example of different samples from CLE images before and after applying the enhancement filter.

Additional experiments are tested by evaluating the model on an individual dataset. The patients' images were split into 60% training and 40% testing. As shown in table 3.4, the model was able to maintain high performance by achieving an overall 93.72%, misclassifying 5 IM as NPL, 4 GM as IM, 3 GM as NPL and 2 NPL as IM.

As a further study, a comparison of the results for the presented model with other state-of-the-art models is demonstrated in Table 3.5. We employ the publicly available dataset provided by the ISBI'16 challenge (*aidasub-clebarrett - Home* 2015) used

Table 3.2: Evaluation of the model with and without (W/O) using the proposed enhancement filter using SVM and Random Forest for model validation. The experiments are tested using LOPO-CV on the 96 patients.

	Image **With** Filter Using SVM classifier				Image **W/O** Filter Using SVM classifier				Image **With** Filter Using RF classifier				Image **W/O** Filter Using RF classifier			
Grade	**IM**	**GM**	**NPL**	**NS**	**IM**	**GM**	**NPL**	**NS**	**IM**	**GM**	**NPL**	**NS**	**IM**	**GM**	**NPL**	**NS**
Sensitivity (%)	**97.0**	90.0	**94.0**	**100.0**	95.0	67.0	85.0	100.0	90.0	**100.0**	86.0	100.0	86.0	88.0	76.0	96.0
Specificity (%)	**96.0**	**99.0**	97.0	**100.0**	79.0	98.0	97.0	99.0	91.0	97.0	**99.0**	100.0	88.0	95.0	92.0	99.0
Precision (%)	**98.0**	**94.0**	83.0	**100.0**	94.0	81.0	**88.0**	97.0	96.0	67.0	55.0	100.0	95.0	62.0	60.0	91.0
F-Measure (%)	**97.0**	**92.0**	**88.0**	**100.0**	96.0	73.0	84.0	98.0	94.0	80.0	67.0	100.0	91.0	73.0	86.0	93.0
Accuracy	**96.05%**				92.01%				91.00%				86.00%			

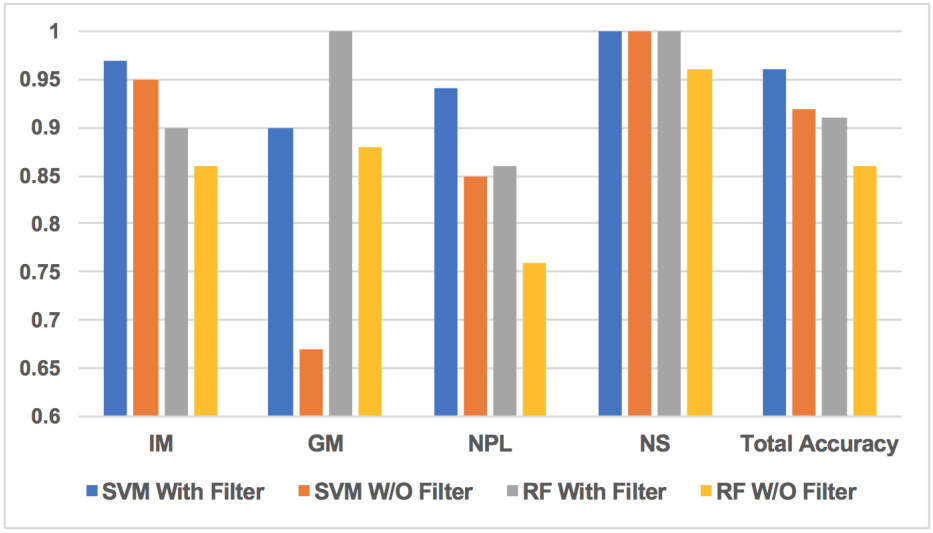


Figure 3.12: Comparison between the accuracy of classifying each grade separately and the overall model with and without applying the enhancement filter to the CLE image using both the SVM and RF classifier.

Table 3.3: Performance measure values obtained after applying different enhancement techniques on the CLE image

	Proposed Filter	HE	Adaptive HE	Median	Wiener	Gaussian
CII	**3.283**	2.458	2.801	648	656	827
Ten.$(\times 10^3)$	**14.712**	13.572	11.903	13.340	11.883	12.849

by both Ghatwary *et al.* (N. Ghatwary, 2017) and the deep learning method by Hong *et al.* (Hong et al., 2017). By comparing the proposed model with Ghatwary *et al.* (N. Ghatwary, 2017), our model surpassed the overall accuracy by 7%. Moreover, by evaluating each class separately a significant improvement was observed in both sensitivity and specificity for the three categories. For the method proposed by Hong *et al.* (Hong et al., 2017), we couldn't compare the results of our model with it. The results illustrated by Hong *et al.* (Hong et al., 2017) in Table 3.5 was based on only a total of 26 images (a small subset from the dataset provided by ISBI'16 challenge (*aidasub-clebarrett - Home* 2015)) from 262 images that are used by our model and Ghatwary *et al.* therefore it was going to be an unfair comparison. Meanwhile, when evaluating the results by Hong *et al.* (Hong et al., 2017), the accuracy showed a low performance of 877%. Moreover, the results indicate that images from GM and NPL were misclassified as IM.

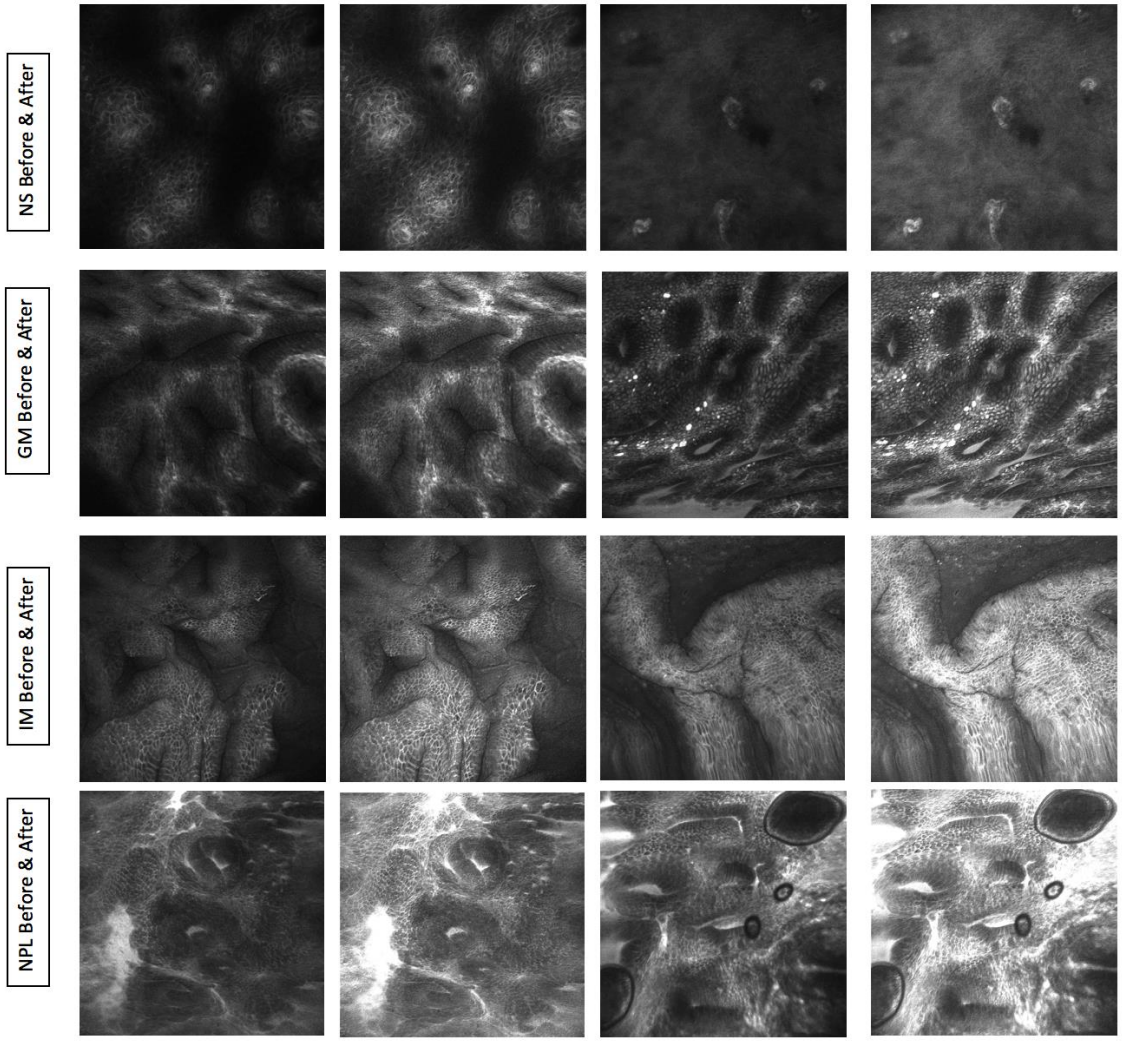


Figure 3.13: Example of different sample of CLE images before and after using the enhancement filter.

Table 3.4: Confusion matrix of the proposed model on an individual dataset, The training set of 60% (58 patients) and testing set of 40% (38 patients)

	IM	**GM**	**NPL**	**NS**	Sensitivity (%)	F-Measure (%)
IM	157	0	5	0	96.0	97.0
GM	4	11	3	0	61.0	78.0
NPL	2	0	19	0	90.0	79.0
NS	0	0	0	22	100.0	100.0
Specificity (%)	90.0	100.0	96.0	100.0	Accuracy = 93.72%	
Precision (%)	96.0	100.0	70.0	100.0		

Table 3.5: Comparison between Proposed Model, Ghatwary *et al.* (N. Ghatwary, 2017) and Hong *et al.* (Hong et al., 2017) Using LOO-CV on 262 Images of Different Stages

	Proposed Model	Ghatwary *et al.*	Hong *et al.*
Total Accuracy (%)			
	97.71	90.46	80.77
Sensitivity (%)			
IM	**98.0**	94.0	100.0
GM	**83.0**	70.0	0.00
NPL	**97.0**	90.0	80.0
Specificity (%)			
IM	**93.0**	88.0	44.0
GM	**100.0**	96.0	100.0
NPL	96.0	**97.0**	100.0

Another comparison assessment is shown in Table 3.6 to illustrate the evaluation of the proposed model against the most recent state-of-the-art methods Veronese *et al.* (Veronese et al., 2013) and Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.), using the same dataset of 336 images with three different classes only (GM, IM, NPL), moreover, using the same evaluation method of a LOO-CV. As shown, the proposed model exceeded the overall accuracy by 2.97% and 18.16% respectively. Also, each class was evaluated separately, beginning with the **IM** class -the main precancerous stage- which is considered the primary target, since its detection through the classification stage is critical to the therapeutic plan. By evaluating the sensitivity, not only the proposed model surpassed (Veronese et al., 2013) by 4% but also, it outperformed (Grisan, Elisa Veronese et al., n.d.) by 22%. However, the specificity of the proposed model falls short by 1% compared to (Veronese et al., 2013). The main reason behind this fall is that one image from the NPL class was classified as IM. On the other hand, in (Veronese et al., 2013) the IM was misclassified as another class; hence their IM specificity was not affected.

GM class is the smallest dataset amongst the three categories in this experimental evaluation. Thus, misclassification of an image leads to an obvious impact on the results. Both sensitivity and specificity of the current model were able to maintain

Table 3.6: Comparison between Proposed Model,Veronese *et al.* (Veronese et al., 2013) and Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.) Using LOO-CV on 262 Images of Different Stages

	Proposed Model	Veronese *et al.*	Grisan *et al.*
Total Accuracy (%)			
	99.11	96.14	80.95
Sensitivity (%)			
IM	**99.0**	95.0	77.0
GM	**100.0**	96.0	78.0
NPL	98.0	**100.0**	**100.0**
Specificity (%)			
IM	99.0	**100.0**	97.0
GM	**100.0**	99.0	94.0
NPL	**99.0**	96.0	84.0

Table 3.7: Comparison of the computation time (in seconds) between Proposed Model, Ghatwary *et al.* (Ghatwary 2017b), Veronese *et al.* (Veronese et al., 2013) and Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.) for image classification

Proposed Model	Ghatwary *et al.*	Veronese *et al.*	Grisan *et al.*
3.9∼6.5	9∼17	7.1∼ 9.2	6.7∼13

the highest performance by correctly classifying all the GM images with no false positives while (Grisan, Elisa Veronese et al., n.d.) and (Veronese et al., 2013) missed 5 and 1 images respectively.

Finally, by evaluating the sensitivity of **NPL** stage, both (Grisan, Elisa Veronese et al., n.d.) and (Veronese et al., 2013) achieved a 100% for this class. However, the proposed model did not experience a significant decline as only a single image was misclassified. On the other hand, the specificity of our model outperformed (Grisan, Elisa Veronese et al., n.d.) and (Veronese et al., 2013) indicating the improvement in decreasing the classification of the other two classes as NPL.

Further investigation has been made since the CLE is an in-vivo process it requires the classification to be done on a real-time basis. Therefore, it is essential to take into account the computation time. The execution time for each phase (*image enhancement* and *feature extraction*) were measured separately. The average processing

time for enhancement of an image required *2-3* seconds while the feature extraction process required *1.9∼2.5 sec* per image. Therefore, the total average processing time required by the proposed model to classify an image is an average of *3.9∼5.5 sec*. We believe that the classification speed could be improved when using a more powerful computer.

Moreover, in Table 3.7 we compare the computation time required to classify a single image using our method with other state-of-the-art methods. As shown, the proposed model was able to classify the stage of the abnormality in less time than the other methods. The average time of the model was faster than Ghatwary *et al.* (N. Ghatwary, 2017) by 7.8 *sec*, Veronese *et al.* (Veronese et al., 2013) by 2.95 *sec* and Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.) by 3.3 *sec*.

One of the main reasons that the methods proposed by Ghatwary *et al.* (N. Ghatwary, 2017), Veronese *et al.* (Veronese et al., 2013) and Grisan *et al.* (Grisan, Elisa Veronese et al., n.d.) take more time is that they are multistage classification models, where the preprocessing, feature extraction and classification is done on several stages based on the abnormality type. Additionally, our model, (N. Ghatwary, 2017) and (Grisan, Elisa Veronese et al., n.d.) are image-based feature extraction systems. Therefore, the processing time towards the feature extraction phase can be considered similar. However, the model in (Veronese et al., 2013) was divided into two phases, a patch-based phase, and an image-based phase which requires more time to divide a single image into patches and select the suitable ones for feature extraction.

The time by the proposed model is considered reasonable and convenient for the examination process since the mean inspection time of the CLE is around 22 minutes. A patient needs between 9 to 45 minutes to be examined, and the CLE image is captured with a rate of 8 frames per second at a resolution of 1024x1024 (Kiesslich, Gossner et al., 2006).

3.6 Summary

A unified classification method is proposed to classify the pathology stages of esophageal abnormality cell deformation stage from CLE images to support the physician's opinion. The automatic classification will help decrease the required biopsy samples and monitor the dysplasia before turning into cancer. Preprocessing steps are first applied to enrich the CLE input image for feature extraction using a novel enhancement filter. The enhancement phase is a vital part of the proposed system, carried out by implementing a preprocessing filter that employs the FI and FD in the sub-bands of the DWT. Subsequently, the FD is applied to the whole image after it regains its original form. Afterward, a proposed multi-scale feature MP-LBP, GLCM, Fractal Analysis, FLBP, and MSER are calculated and fed into two classifiers the SVM and RF.

The experimental results show that the proposed method achieves promising results in classifying the CLE images into the cell deformation stage. Applying the enhancement feature helped improve the classification results when compared to the original images. Additionally, selecting suitable features that fit the properties of the stages leads to higher performance. The proposed system was able to achieve state-of-the-art results with an overall accuracy of 96.05%.

This work has been published in the Journal of Medical Imaging (JMI). Additionally, the preliminary model and results were published in the Conference of SPIE Medical Imaging and the Poster presentation was awarded the "CUM Laude Award" for being the best poster presentation (Certificate is available in Appendix B). Moreover, a challenge was held under the title "Esophagus microendoscopy images in Barrett's surveillance", the challenge requested a model to classify between the three different stages of esophagus cell deformation. The initial results of the proposed model were the winner of the "Esophagus microendoscopy images in Barrett's surveillance" challenge, which was announced in the IEEE ISBI'16 on 13th April 2016 at Prague (Winning Certificate is available in Appendix B). The CLE tool is used to capture zoomed histopathological images for the esophageal abnormalities detected by

the physician using the WLE and HD-WLE. The next chapter will investigate the detection of esophageal abnormalities from the endoscopic images.

Chapter 4

Esophageal Abnormality Detection from Endoscopic Images using Deep Learning

4.1 Introduction

Most of the esophageal abnormality detection methods presented in literature relied on extracting handcrafted features (i.e. (Van Der Sommen, F. Zinger S. et al., 2014) and (L. Souza et al., 2017)). However, the selection of the appropriate handcrafted features is challenging as it should be chosen according to the characteristics of the image to provide a better description of the abnormal area. Furthermore, many experiments and optimization should be performed to obtain the optimal parameters for feature extraction and designing an optimal classifier.

Deep learning has been widely applied in the medical image detection and classification field by extracting features through convolutional neural networks (CNNs) (Greenspan, Van Ginneken and Summers, 2016). Deep CNNs generate features from the images through learning from the dataset, increasing its generalization and scalability for automatic detection (Yi et al., 2017). Recently, a few approaches have been suggested to improve the performance of the automatic detection of abnormal esophageal regions. The most recent method was proposed by Mendel *et al.* (Mendel et al., 2017). They suggested extracting CNN features from non-overlapping patches using ResNet (He et al., 2016) based on **transfer learning** (i.e from non-medical

domain). The method was tested to detect the EAC region only on a small-sized dataset.

In literature, different CNN architectures are constructed to learn and provide informative features for the detection and classification methods such as: (*AlexNet (Krizhevsky, Sutskever and Hinton, 2012), VGG'16 (Simonyan and Zisserman, 2014), ResNets (He et al., 2016), etc.*). The depth of the CNN network shows a significant impact on the performance of the network but getting deeper without changing in the structure can lead to poor performance, loss of information and facing vanishing the gradient parameter (Wenqi Liu and Zeng, 2018). To overcome these problems, Huang *et al.* (G. Huang et al., 2017) introduced the Densely Connected Convolutional Networks (DenseNet). The advantage of DenseNet architecture is that it lowers the number of parameters, improves the gradient and information flow throughout the network which makes it easier to train. Additionally, DenseNet encourages feature reuse by connecting the output of each layer to another layer.

Recently, the combination of handcrafted features with CNN features showed that it can boost the performance of the model (Hosseini, S. H. Lee and N. I. Cho, 2018). Texture features such as Gabor features has shown its effectiveness when merged with CNN features by providing low-level texture information (Shi et al., 2018). The advantage of merging both sets of features have been confirmed in different studies (Luan et al., 2018; Yao et al., 2016; Kwolek, 2005; Y. Chen et al., 2017). Gabor filters have been known for strengthening the texture details provided through spatial information. Additionally, concerning the esophageal abnormality detection, the Gabor features have shown its efficiency in detecting the intestinal juices (Iorio et al., 2006).

In this chapter, we provide a general overview of deep neural network models focusing on CNN's. Then we present an overview of state-of-the-art abnormality detection methods (i.e. from images) in the field of supervised-handcrafted based methods and deep learning-based methods. Next, we investigate the capability of different deep learning object detection methods to detect different esophageal abnormalities from endoscopic images. Then we propose two novel methods that depend on deep

learning to accurately and effectively find abnormal regions. In the first model, we study the incorporation of Gabor handcrafted features with CNN features to improve the detection performance. In the second model, we propose an innovative deep learning model that has more than one network to extract CNN features from the original and a generated Gabor Fractal image that will boost the overall performance. The contributions of this chapter can be listed as follows:

- Different state-of-the-art CNN based detection methods such as *R-CNN, Fast R-CNN, Faster R-CNN*, and *SSD* have been adapted and evaluated to automatically identify esophageal abnormality regions from endoscopic images.
- A novel unified framework is presented based on hybrid features that combine information from deep learning and handcrafted features to automatically detect esophageal abnormalities from endoscopic images. Our method integrates the DenseNet features with Gabor handcrafted features into the detection framework.
- A novel Gabor Fractal DenseNet Faster R-CNN (GFD Faster R-CNN) is proposed which is a two-input network adapted from the Faster R-CNN to address the challenges of esophageal abnormality automatic detection.
- The proposed frameworks are trained end-to-end and extensively evaluated on the available datasets (Kvasir and MICCAI'15 as mention in Chapter 2, Sec. 2.5) with the different types of abnormalities.

4.2 Overview of Deep Neural Network Models

Deep Learning is a subfield of machine learning methods that uses deep neural networks. The deep networks can generate features from the images through learning from the dataset, increasing its generalization and scalability for automatic detection and classification (Jin Liu et al., 2018). Lately, deep learning has been widely applied in the medical image detection and classification field by extracting features through network architectures specially CNN (Litjens et al., 2017).

4.2.1 Introduction To CNN

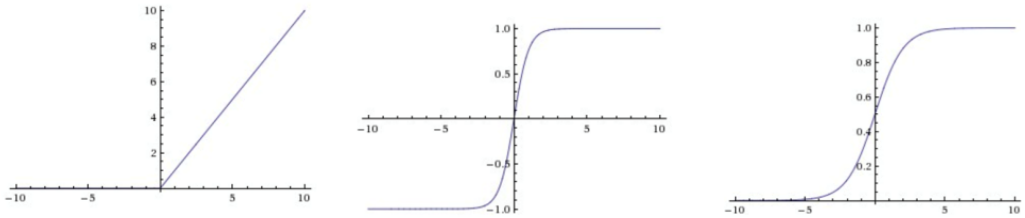
CNN is a supervised learning model that analyzes the input data in a feed-forward manner. The CNN has shown to have an accountable performance when dealing with grid-like topologies as images and videos (Weibo Liu et al., 2017). The main target of CNN is to learn high-order features within the data through convolutions.

The standard CNN architecture for feature extraction is composed of a series of layers that allow extracting a set of discriminative features at different levels (Greenspan, Van Ginneken and Summers, 2016). The CNN main layers are; an input layer, convolutional layer, pooling layer, activation layer, and fully connected layer. Each layer of CNN is explained as follows:

- *Input Layer:* The input layer holds the raw pixel values of the input data that will be exposed to the network. In the case of the endoscopic data; the width and height of the input layer are the spatial dimensions of a single frame and the third dimension represents the color channels. For the video, a fourth dimension is presented for the number of frames or sequences per video.

- *Convolutional Layer:* The core layer of a CNN is the Convolution Layer (Conv) which is responsible for most of the computational learning operation throughout the network. A convolutional layer includes a set of filters whose parameters need to be learned. The size of the filters (i.e. height and weight) is smaller than the volume of the input. Each filter is slid over the input volume and the result is a filter map holding the dot product result computed at every spatial location of the input. Each Conv in the network is composed of a set of filters that produce a feature map. Convolution preserves the relation between pixels by learning image features using these filters. The output volume of the convolutional layer is obtained by stacking the activation maps of all filters along the depth dimension. The output of the convolutional layer is the feature map.

  The sliding operation for a filter is named **stride**. Stride represents the number of pixels shifted over the input volume. For example, if the stride= 1 then the



(a) ReLu (b) Tanh (c) Sigmoid

Figure 4.1: Activation Functions

filter is moved one pixel at a time while if stride= 2 then filter is moved two pixels at a time and so on.

A **padding** operation might be required if the filter dimension does not fit the input dimensions. To make them similar, the input of spatial margins is padded with zeros (i.e. know as zero-padding).

- *Activation Layer:* The activation layer is responsible to convert the outputs from the convolutional layer to an output matrix that can be used by the following layer. A nonlinear function $f(x)$ is used to get the output of the layer using the inputs and corresponding weights. Specifically, it maps the results to values between a certain range $a < x < b$ where a & b are based on the function. In this thesis, we utilize three of the popular activation functions which are ReLu, Sigmoid (σ) and Tanh that are illustrated in Fig. 4.1 and represented by:

$$ReLu(x) = max(x, 0) \tag{4.1}$$

$$\sigma(x) = \frac{1}{1 + \exp^{-x}} \tag{4.2}$$

$$Tanh(x) = \frac{\exp^{x} - \exp^{-x}}{\exp^{x} + \exp^{-x}} \tag{4.3}$$

In the literature, the ReLu is the most common activation function used in deep learning methods as it can learn fast in large networks.

- *Pooling Layer:* The pooling layer reduces the complexity of the network by decreasing the number of parameters and computation of a network consequently it controls overfitting and leads to more robust features. It acts as a down-

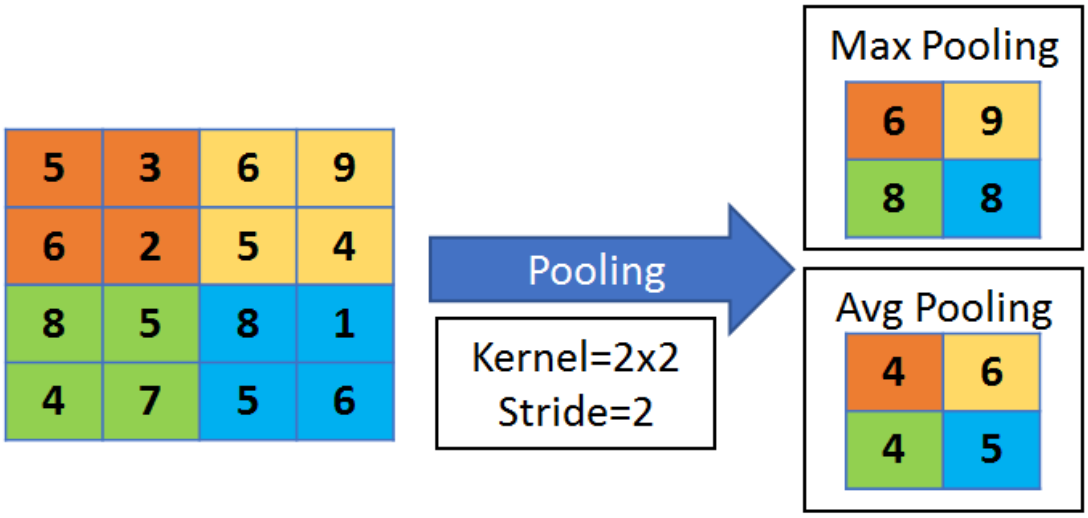


Figure 4.2: An example of the output from Max Pooling and Avg. Pooling for the same location with kernel size 2×2 and stride=2.

sampling layer along the spatial dimension and commonly applied when a number of filters are operated on the previous set of Conv layers.

Pooling layers perform similar to Conv layers, but they operate a particular function. The common pooling methods are: *Maximum Pooling (Max. Pooling)* that select the maximum value in a certain filter region and *Average Pooling (Avg. Pooling)* that calculates the average value in a filter region. Figure 4.2 represents an example for both max-pooling and avg-pooling with kernel size= 2×2 and stride=2.

- *Fully Connected Layer (FC):* In this layer, all neurons in a layer are connected to every output from previous layers. FC can learn weights that can identify a candidate class, therefore, it is responsible for the classification of the classes of the data using the extracted feature map.

4.2.2 Commonly Used CNN architectures

The main layers of a CNN (as discussed in the previous section) are used to construct networks for different purposes. Different CNN architectures are built to learn and provide informative features for the detection and classification methods. In the

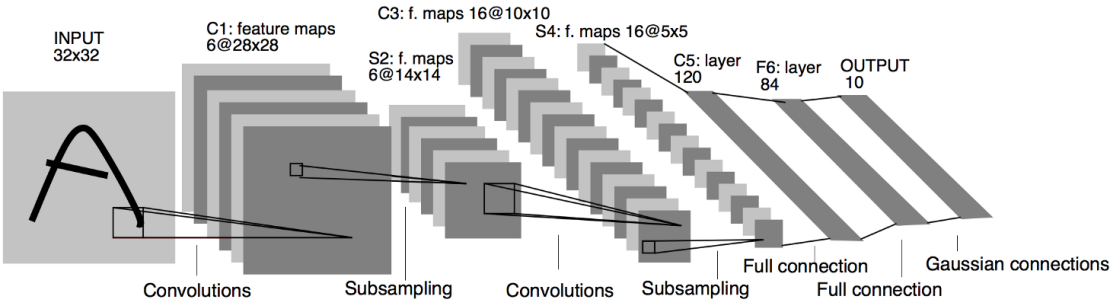


Figure 4.3: Illustration of the LeNet-5 architecture for digit recognition proposed by (LeCun et al., 1998)

following subsection, we will explain the most common CNN architecture that has been widely used.

- **LeNet**

  LeCun et al. (LeCun et al., 1998) introduced a small straightforward network called LeNet to classify handwritten digit numbers (i.e. MNIST dataset). The network architecture was composed of two sets of convolutional layers, ReLu activation and average pooling layers followed by two fully connected layers with an activation function in between. Figure 4.3 illustrates the architecture of LeNet-5.

- **AlexNet**

  Krizhevsky et al. (Krizhevsky, Sutskever and Hinton, 2012) proposed a CNN architecture named AlexNet which is very popular in the tasks related to computer vision. AlexNet is an advanced version of LeNet but with deeper, bigger and stacked Convolutional layers on top of each other as shown in Fig. 4.4. The network is composed of eight layers in total (five Conv and three Fully connected layers). The Alexnet architecture is the winner of Image Net ILSVRC challenge 2012 (*Large Scale Visual Recognition Challenge 2014* 2014).

- **VGGNet**

  Simonyan and Zisserman (Simonyan and Zisserman, 2014) introduced the VGG network that was a runner up in ILSVRC 2014 (*Large Scale Visual Recognition Challenge 2014* 2014). They suggested that the performance of a model can be improved by increasing the depth of the network. A small window kernel for

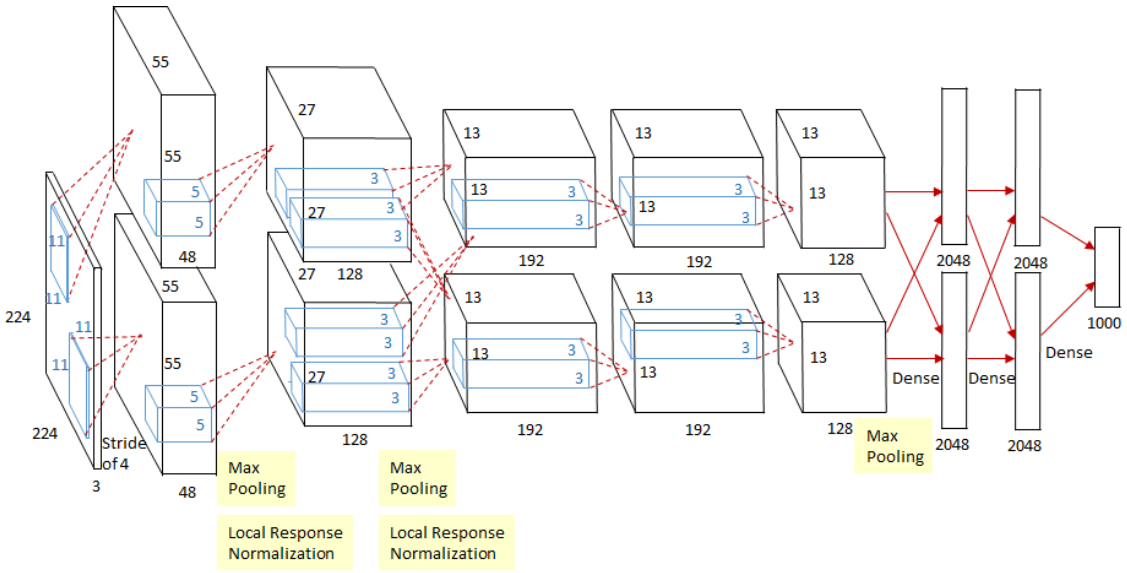


Figure 4.4: Illustration of the AlexNet architecture for image classification proposed in (Krizhevsky, Sutskever and Hinton, 2012)

the Conv (3×3) and max-pooling (2×2) was used throughout the network from beginning to end which lead to a high number of trained parameters. Different architectures for the VGG network were proposed by varying the number of layers (i.e. 11, 16 and 19). The VGG'16 was recommended to have the best performance compared to the computation complexity with a total of 16 layers as shown in Fig. 4.5.

- **GoogleNet**

  Szegedy et al. (Szegedy, Wei Liu et al., 2015) introduced the GoogleNet architecture that is composed of 22 layers in the network. The main addition they introduced to the network is an *Inception Module* that reduced the number of parameters in the network. The inception module; acts as a multi-level feature extraction by applying multiple convolution filters for the same input and concatenating the results.

- **ResNet**

  He et al.(He et al., 2016) developed the residual networks (ResNet) architectures. They proposed the concept of shortcut connection (i.e. know as skip connection) that skips the training of few layers. Moreover, they incorporate the use of Batch Normalization (BN) after each Conv. Additionally, they re-

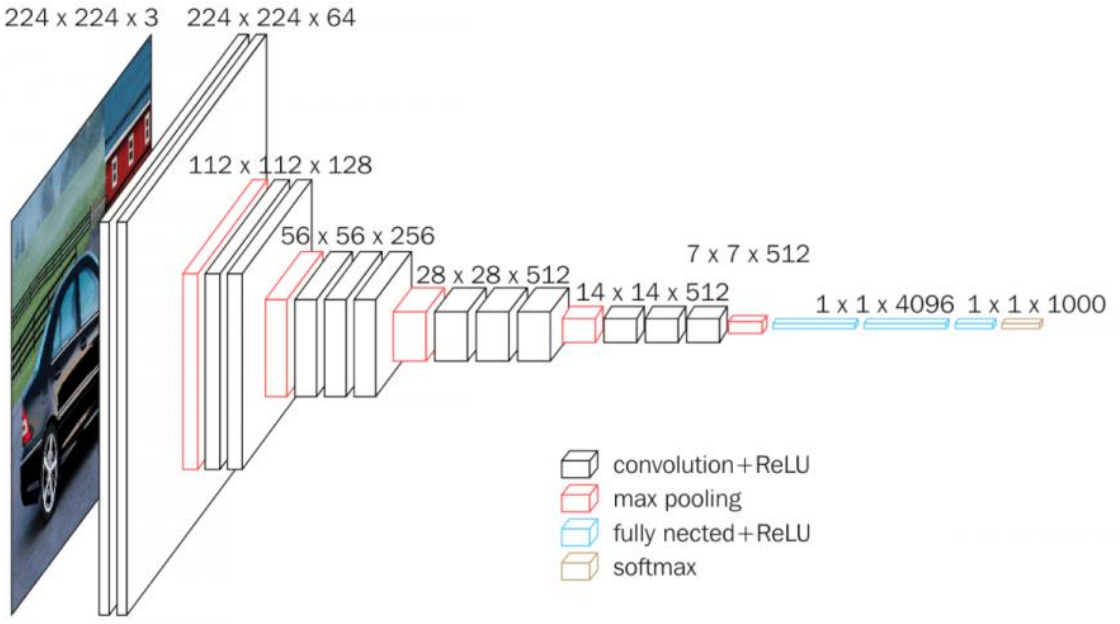


Figure 4.5: Illustration of the VGG'16 architecture for image classification proposed in (Simonyan and Zisserman, 2014)

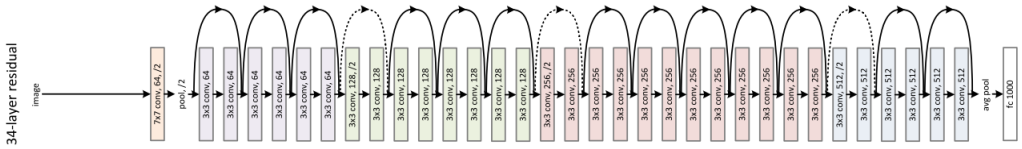


Figure 4.6: Illustration of the 34 layer ResNet as proposed by (He et al., 2016)

move the fully connected layer from the end of the network. Despite its good performance, the ResNet is regarded as one of the very deep networks that include a large number of trained parameters.

- **DenseNet**

  Huang et al.(G. Huang et al., 2017) presented the concept of DenseNet architectures. DenseNet encourages feature reuse by connecting the output of each layer to another layer as shown in Fig. 4.7. The network composed of different components such as DenseBlock, Composite Layer, Transition Layer, and Growth Rate. *In this thesis, we adopt the DenseNet to build our abnormality detection model from images and videos. More details about DesneNets will be explained in Chapter 5* (Section 4.5.1).

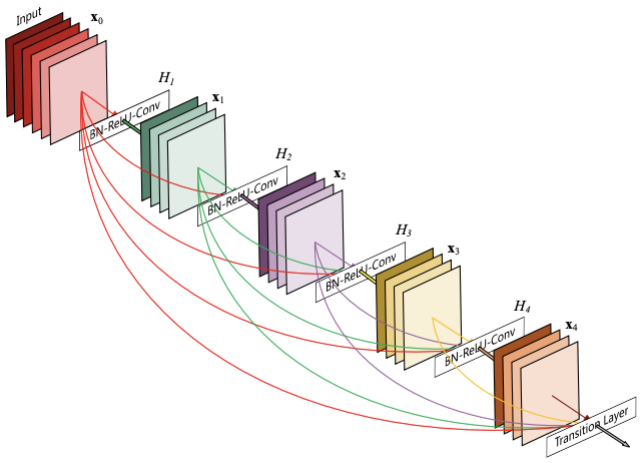


Figure 4.7: Demonstration of a 5-layer dense block. Each layer uses all previous feature-maps as input (G. Huang et al., 2017).

4.3 Overview of the Current State-of-The-Art Detection from Image Methods

In this section, we review the different techniques for esophageal abnormality detection based on the type of endoscopy used for examining the infected esophagus. The focus is on the key methods that have been reported recently in the literature based supervised methods with handcrafted features and CNN deep learning methods .

4.3.1 Supervised methods with handcrafted features for esophageal abnormality detection

Automatic detection of cancerous regions using WLE is presented by Yamaguchi *et. al* (Yamaguchi, Yoneyama and Minamoto, 2015) that took advantage of fractal dimension properties to apply the detection process. As a first step, the image is decomposed into four components *Red, Green, Blue* and *Luminance*. Afterwards, regions of the image that clearly doesn't have cancer are clipped out to save more processing time (i.e. aiming to reach a real-time model). Followed by standardizing the size for processed images by resizing them to 1024×1024. Later, the images are decomposed into its DWT form, and only the low-layer component is utilized and

divied it into small non-overlapping blocks of size 128x128. More blocks are discarded that have a luminance value that is less than a total average value of luminance of all blocks. The remaining qualified blocks are exposed to DWT twice and divided into smaller sub-blocks. Each of these steps is applied to every component layer that was decomposed in the first step. The feature vector extracted for the classification phase is finally calculated by multiplying the fractal dimension of each component layer by using the box-counting method extracted from for each sub-block. The block region is considered classified as a cancer region if it has a very low fractal dimension value. The problem with this method that it is very time-consuming where a single image can undergo 3 minutes to reach a decision.

Matsunage *et. al* (Ohura et al., 2016) grabbed attention to the abnormal regions of early esophageal cancer after normalizing the input image to a certain range. Images are then converted from RGB to HSV color space. Later, the Dyadic Wavelet Transform (DYWT) is then applied to the S and V components and their low-level frequency sub-band are fused together. Following on, contrast enhancement is applied to that fused image which is divided afterwards into 16x16 non-overlapping blocks. The sum of the fractal dimension value is calculated as in their previously proposed work (Yamaguchi, Yoneyama and Minamoto, 2015) to distinguish if the blocks are normal or abnormal.

In all the WLE models/techniques discussed above, the evaluation was done through visual/qualitative approach only. The visual evaluation is done by visually comparing the detected region, by the proposed methods, with the annotation done by the experts (as a Ground-Truth). Hence, no quantitative evaluation results, through the common evaluation measures, were given.

In order to implement a CAD system of early cancer detection in the esophagus, a study was represented by Setio *et. al* (Setio et al., 2013) that evaluated the various texture features extracted from HD-WLE images. The proposed study assessed the efficiency of Local Binary Pattern (LBP), Texture Spectrum, Histogram of Oriented Gradients (HOG), Dominant Neighbor Structure (DNS), Grey Level Co-occurrence Matrix (GLCM), Fourier feature and Gabor Features. After discarding the irrelevant

texture tiles from the images as a preprocessing phase, the features were extracted. The results concluded that merging between the Gabor features and the Color features achieved 96.48% compared to the baseline of annotated accuracy and against the combination of other features. The method utilized the Principal Component Analysis(PCA) for reducing the dimension of the features and were classified using the SVM.

Based on this conclusion an automatic detection model has been proposed by Sommen *et. al* (Van Der Sommen, Svitlana Zinger, E. J. Schoon et al., 2013) (Van Der Sommen, F. Zinger S. et al., 2014). The chosen features were used to detect and annotate infected lesion in the esophagus. The implemented method extracts the desired features and classifies them using SVM to allocate the region of interest. The dataset used consisted of 32 images from 7 different patients. Comparing the results with the specialist annotation the system was able to achieve 85.7% with a recall of 95% and precision 75%. The model needed to increase its robustness and to have the ability to be real-time.

Later on, the previously stated method was extended in (Sommen et al., 2016) to automatically annotate the neoplastic lesions in Barrett's esophagus and compared to the annotations of 5 experts. The proposed method was tested on 100 images from 44 patients. The analysis of the proposed model was able to accomplish a sensitivity of 86% and a specificity of 87% . The results achieved were almost the same or less in comparison with the experts in the annotation and detection as it is considered the ground truth. As a result, the study was considered a promising start where it was extended in (Janse et al., 2016) by changing the classifier from SVM to Random Forest to benefit from evaluate the classifier efficiency. The replacement of the classifier improved the results by 6% and 11% reaching a recall of 90% and precision of 75%.

Souza Jr. *et al.* (L. Souza et al., 2017) proposed an investigation of the feasibility of the SVM to classify lesions in Barrett's esophagus based on Speeded Up Robust Features (SURF) descriptors (Bay, Tuytelaars and Van Gool, 2006). Two experiments were carried out by extracting the Surf features from the full image and another

from the abnormal region (i.e. using the EAC ground truth regions annotated by experts). The results based on full image analysis showed a sensitivity of 77% and specificity of 82% while the abnormal region-based approach has a sensitivity of 89% and specificity of 95%. These results were analyzed based on the LOPO-CV approach and SVM classifier. Afterwards, Souza Jr. *et al.* (De Souza et al., 2017) proposed an Optimum-Path Forest (OPF) classifier to identify BE and adenocarcinoma from HD-WLE images. Features were extracted from the images using the Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) and the SURF to design a bag-of-visual-words (BoW) to be an input for the OPF and SVM classifiers. Results showed that the OPF outperformed the SVM with sensitivity of 73.2% (SURF) - 73.5% (SIFT), specificity of 78.2% (SURF) - 80.6% (SIFT), and accuracy of 73.8% (SURF) - 73.2% (SIFT).

Boschetto *et al.* (Boschetto, Gambaretto and Grisan, 2016) suggested an automatic classification method that differentiates between normal and metaplasia (abnormal) regions. The method employed the superpixel method to cluster the image into regions by using Simpler Linear Interactive Clustering (SLIC) (Achanta et al., 2012). SLIC depends on K-mean clustering method to benefit from the simplicity and efficiency of computation time. Later on, eight different features are extracted from each superpixel. These features were divided as the mean of intensity values for the color channel of an image, the mean intensity values were extracted again, but after applying three different filters which are Entropy filter, Range filter, and Top-hat filter. The last two values added to the feature vector were the contrast and homogeneity that are calculated from the GLCM texture method. Random forest classifier was used to classify between normal and metaplasia lesion from 116 NBI images of different patients. The method achieved an overall accuracy of 83.9% accompanied with a sensitivity of 72.2% and specificity of 87.3%. Since the method was mainly proposed as a proceeding step before classifying the type of metaplasia, therefore, the accuracy of the model needs to be improved.

A study was proposed by Kage *et al.* (Kage et al., 2009) using NBI endoscopy images to prove the efficiency of employing automatic detection by classification. The model extracted selected features from 326 Region-of-Interest (ROI) that were annotated by

experts and classified as epithelium, cardiac mucosa, and Barrett's esophagus (BE). The feature vector in the proposed work was composed Co-occurrence matrices, Sum and Difference of histogram, Statistical geometrical, Gabor Filter. Later, a forward selection approach was used to reduce the high dimensional feature vector size. The evaluation of this study measured the performance of each selected feature separately and also by combining them altogether by using the Euclidian distance as a similarity metric. Accuracy results ranged between 85% and 92%. The best accuracy of classifying for the BE individually was only 74%.

Rajan *et al.* (Rajan et al., 2009) applied several experiments using different conventional classifiers: *SVM, KNN*, and *Boosting* on images from various endoscopy modalities: *WLE, NBI* and *Chromoendoscopy.* Features were selected based on a study proposed by Munzenmayer (Münzenmayer, 2006) for color and texture analysis of medical images. The dataset used for the evaluation of the model was divided into four categories divided as Normal Squamous, Gastric Mucosa, BE and High-grade dysplasia. By down-sampling, the endoscopic image and extracting features as suggested by (Münzenmayer, 2006) the images were classified as one of the four types. After testing the different classifiers, the accuracy for detecting BE varied from 36.36% up to 89.17% the classifier used.

Klopm *et al.* (Klomp et al., 2017) (Swager et al., 2017) studied the prospect of computer-aided systems to automatically detect the presence of dysplastic tissues in esophageal VLE images. A set of new features derived from standard GLCM is suggested based on the clinical prediction model (i.e. Irregular glandular structures, Surface maturation, and Layering within the tissue) to identify dysplastic regions. Using the SVM classifier on a dataset of 60 VLE images (*30 dysplastic* and 30 non-dysplastic) the model was able to achieve 0.95 AUC value compared to 0.81 gained from the clinical model.

Scheeve *et al.* (Scheeve et al., 2019) suggested a new gland-based image feature named "*gland statistics*" that merges texture and geometry analysis to classify 122 VLE images gathered from 18 BE patients with and without early BE neoplasia. The feature vector is extracted from first-order statistics: *mean, standard deviation,*

minimum, maximum, skewness, kurtosis, energy, and *entropy*. Also, the convexity, solidity, and disperse from the segmentation masks of the glands for the geometric representation were extracted. Different 8 classifiers were tested that showed an average AUC value of 0.88 with the best performance of using Linear SVM with 91%.

4.3.2 CNN methods for esophageal abnormality detection

Recently, CNN based methods have started to draw attention to EAC detection through transfer learning. Transfer learning is the process of initializing the weights of the suggested network from a pre-trained model of a different or non-medical domain. Mendel *et al.* (Mendel et al., 2017) studied the analysis of BE using CNN to classify patches in an HD-WLE image into cancerous and non-cancerous from MICCAI'15 dataset (i.e. 100 images). Regarding the experiments, the image was first divided into non-overlapping 224×224 patches and sampled as cancerous and non-cancerous based on a certain threshold t. Each patch has an output probability that was compared to the value t to decided if it is a cancerous region or not. The deep residual network (ResNet) (He et al., 2016) was used as the deep learning method for feature extraction and classification from each patch. After testing the performance of classification at seven different values for threshold t, the best performance was achieved at $t = 0.8$ resulting in a sensitivity of 94%, specificity of 88% and F-measure of 91%.

This model has been later extended by Ebigbo *et al.* (Ebigbo, Mendel et al., 2019), where more datasets from different endoscopic modalities have been examined and proposing a model for segmenting the abnormal region. A dataset named *Augrburg dataset* composed of 148 images gathered from WLE and NBI endoscopy has been evaluated along with the MICCAI'15 dataset. Using weights from ResNet, the detection results for the WLE images showed a sensitivity of 97% and specificity 88% while for the NBI images the sensitivity achieved is 94% and specificity of 80%.

Moreover, Reil *et al.* (Van Riel et al., 2018) proposed an early EAC detection method using transfer learning. The idea of the model is to extract intermediate

CNN features of the state-of-art CNN network and classify them using the standard conventional classifiers (SVM and RF). Different architecture, such as *AlexNet (Krizhevsky, Sutskever and Hinton, 2012), VGG'16 (Simonyan and Zisserman, 2014)* and *GoogleNet (Szegedy, Wei Liu et al., 2015)* were evaluated with the weights transferred from the non-medical domain of *ImageNet* using both classifiers individually. After evaluating all the networks, the best performance was achieved by AlexNet-SVM with 0.92 area-under-the-curve (AUC) value.

Putten *et al.* (Putten et al., 2019) suggested a model to automatically identify informative frames to provide an easier analysis for non-expert endoscopists to examine BE abnormalities. The method used a CNN pre-trained model that is composed of 18 layers similar to ResNet to classify the frames and combined the network with a Hidden Markov Model (HMM) that uses the temporal information for improved classification. By adding the HMM to the CNN network the sensitivity results were improved by 10% when compared to CNN only. Later on, Struyvenberg *et al.* (Struyvenberg et al., 2019), customized a hybrid ResNet-Unet architecture to automatically characterize BE neoplasia from NBI-zoomed images. The suggested network was trained on three stages using three different datasets. Firstly, the model was pre-trained on a different dataset of 494,364 labeled endoscopic images named "GastroNet". Aftewards, the model was trained using WLE images composed of 690 BE neoplasia and 557 non-dysplasia BE (NDBE). Lastly, the model was trained and tested through both transfer and ensemble learning techniques using the third dataset of NBI-zoomed images. The results of the CAD system showed an average AUC of 91%, accuracy of 84%, the sensitivity of 88% and specificity of 78% to correctly differentiate between NDBE and BE neoplasia.

4.4 Overview of Deep Learning esophageal abnormality detection methods from endoscopic images

In this section, we take advantage of the recent development in object detection methods that utilize CNNs to locate esophageal abnormalities in endoscopic images by employing the state-of-art CNN methods and adapting them to our dataset.

There exist various object detection methods that rely on CNN features for final detection which are divided into two categories: Two-stage and One-stage detector. Two-stage detector methods include Regional-Based Convolutional Neural Network (R-CNN) (Girshick et al., 2015), Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2015). One-stage detector methods include Single-Shot Multibox Detector (SSD) (Wei Liu et al., 2016) where these types of methods suggest predicted output directly from an image without region proposal stage. Each of these methods will be described in detail in the following subsections.

R-CNN

Girshick *et al.* (Girshick et al., 2015) first proposed a regional-based convolutional neural network (R-CNN) as a leading framework for general object detection method using deep learning. The R-CNN method is composed of three main steps as shown in Fig. 4.8. First, the input image is scanned to generate over 2000 region proposals that might contain an abnormal region based on a selective search algorithm (Uijlings et al., 2013). The goal of the selective search algorithm is to provide several candidate regions that belong to an abnormality. It starts by generating an initial sub-segmentation to find a small set of independent class objects. Then it keeps repeating combining similar regions into larger ones using the greedy algorithm to find the most similar ones. Finally, outputs candidate regions called proposals that contain abnormality. After that, CNN is run over each of the proposals to extract features of this region. Finally, the extracted features from the previous step are fed into an SVM classifier to classify this region into a suspected abnormality and a

Linear regressor is used to refine the bounding box if the object exists. The method merged between the original region proposal methods with CNNs, but it was considered slow for real-time processing and computationally expensive in the training process.

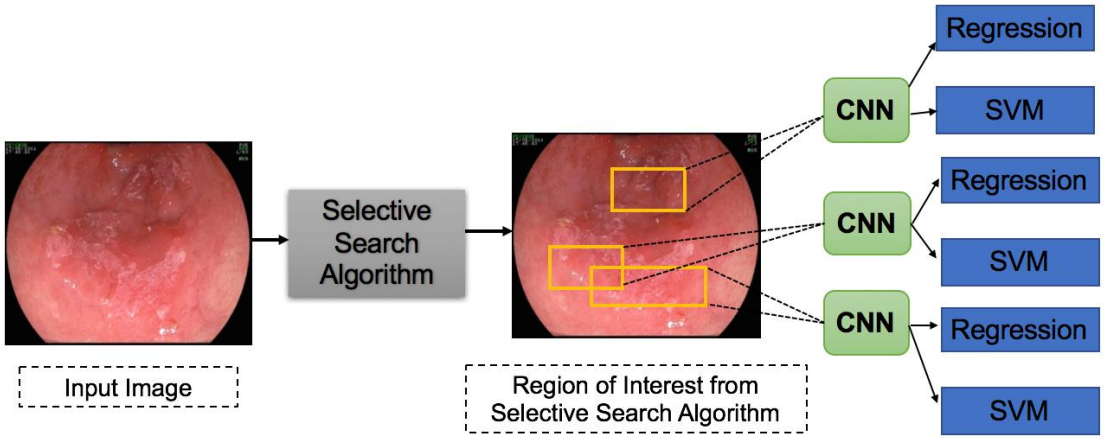


Figure 4.8: General architecture of the **R-CNN**. The selective search algorithm is first applied to find abnormal candidate regions. The SVM is then used to classify the class based on the feature map from the CNN applied to candidate regions, and the linear regression is used to adjust the bounding box location.

Fast R-CNN

To overcome the R-CNN drawbacks, Girshick proposed the Fast R-CNN (Girshick, 2015) through two main modifications. Firstly, the CNN feature extraction is performed over the image itself rather than over the proposed regions. Therefore, the generated region proposals are based on the last feature map of the network, and CNN is only trained once on the full image. Secondly, the SVM classifier is replaced with a single softmax layer that outputs a class probability instead of running multiple SVMs for various object classes. Additionally, an ROI pooling layer is added to the last convolutional layer to unify the feature vector size before applying the softmax classification. The performance of the Fast R-CNN was improved regarding the speed compared to the R-CNN, but the executed selective search algorithm still caused a considerable overhead. The architecture of the Fast R-CNN is illustrated in Fig. 4.9.

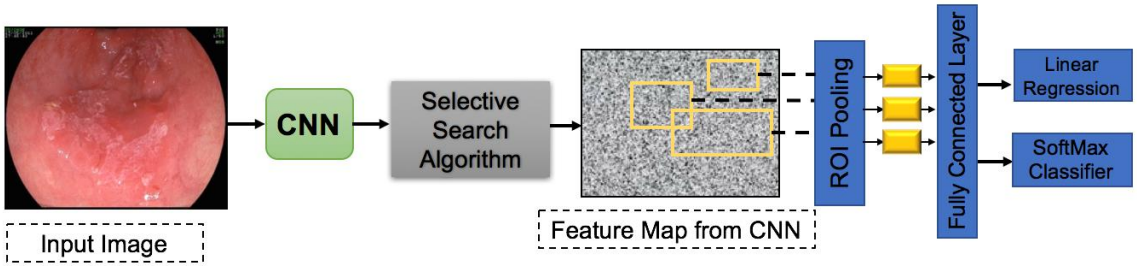


Figure 4.9: General architecture of the **Fast R-CNN**. The CNN is applied to the input image to extract the feature map and the selective search algorithm is performed to find abnormal candidate regions. The ROI is applied after that to unify the feature vector size for classification using Softmax classifier.

Faster R-CNN

Ren *et al.* (Ren et al., 2015), suggested combining a proposed Region Proposal Network (RPN) instead of the selective search into the Fast R-CNN leading to a more real-time method called Faster R-CNN. The proposed RPN generates region proposals for each location using the last feature map produced from the CNN based on *anchor boxes*. The anchor boxes are detection boxes that have different sizes and ratios that are compared to the ground-truth during the training process. For each location in the feature map, there are K different anchor boxes centered around it as shown in Fig. 4.10. The total number of anchor boxes per image is $(K \times W \times H)$ where the W and H are the sizes of the last feature map. During training, each generated anchor box is compared to the ground truth object location. Boxes that overlap the groundtruth with an *Intersection over Union (IoU)* based on a certain threshold is considered as an object (no class specified). The IoU is calculated as follows:

$$IoU = \frac{A_{gt} \cap A_p}{A_{gt} \cup A_p} \tag{4.4}$$

where, A_{gt} is the area of the ground truth bounding box while A_p is the predicted bounding box from the regression layer. The selected anchor boxes are passed on as region proposals from the RPN stage with a classification score for each box and four coordinates that represent the location of this object. Some region proposals highly overlap each other, therefore *non-maximum suppression (NMS)* is used to

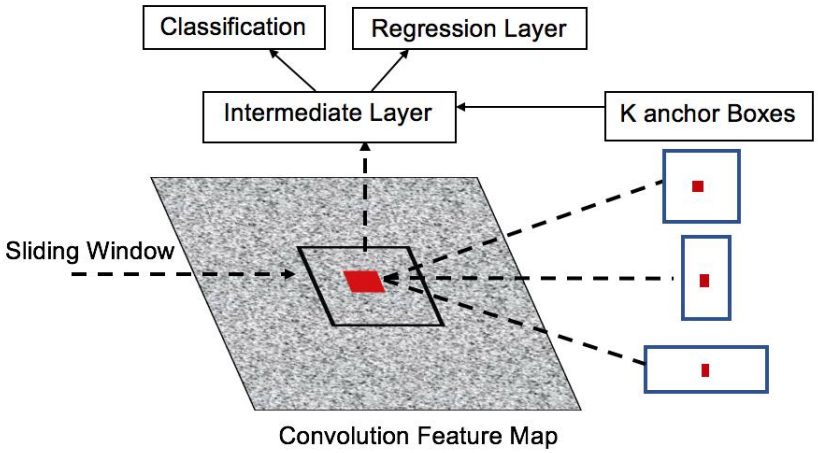


Figure 4.10: An example of different anchor boxes with different sizes and ratios for a specific location in the RPN stage.

prune the redundant regions leading to a reduced number of region proposals. Later on, the selected region proposals are fed into the next phase as in Fast R-CNN. The ROI pooling divides the input feature map from candidate anchor boxes into a fixed number of almost equal regions. Max-pooling is applied to these regions; consequently, the output from the phase is always fixed size regardless of the input size. The general architecture of the Faster R-CNN method is shown in Fig. 4.11.

Single Shot Multibox Detector (SSD)

Liu *et al.* (Wei Liu et al., 2016) presented the SSD) method. The SSD is considered a faster deep learning object detection method compared to previously discussed methods as it generates the predicting bounding box and classifies the object within it in a single operation while processing the image. During the training process, the SSD takes the image and the ground-truth as inputs. Following that, the image is passed through a series of convolutional layers that are combined throughout the network as shown in Fig. 4.12. The SSD generates a list of bounding boxes for each location using priors (i.e. same as anchors in Faster R-CNN) and then adjusts it to be close to the ground truth location as much as possible. Although the number of generated boxes from SSD is considered a huge number compared to the other methods it does not guarantee to have an object inside it. An NMS is applied to minimize the number of boxes by grouping the highly overlapping regions

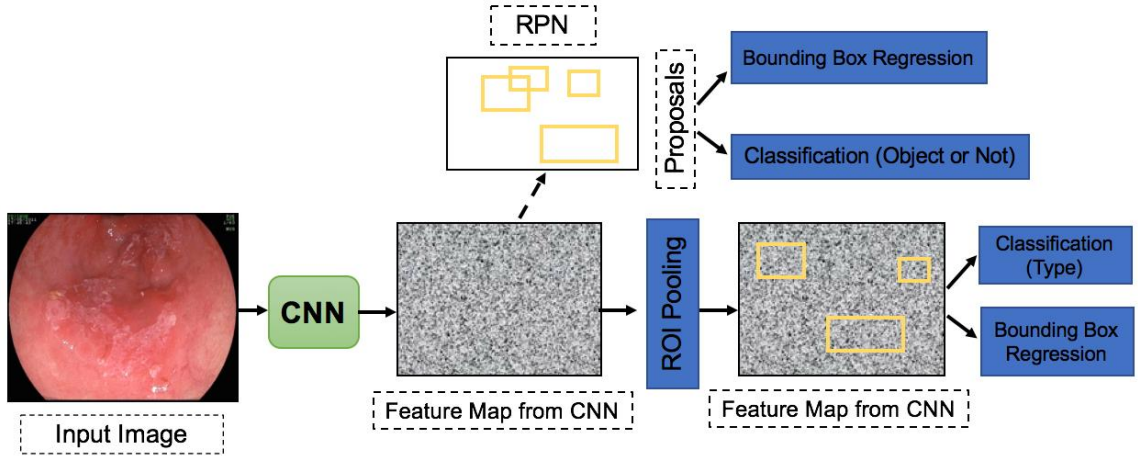


Figure 4.11: General architecture of the **Faster R-CNN**. The CNN is applied to the input image to extract the feature map that is later used by both the RPN and the ROI pooling layers (feature map is shared between both). The RPN outputs the classification score and bounding box location of the candidate region proposals that are passed on to the next stage. The ROI layer unifies the feature vector size of the candidate region proposal that is classified using softmax.

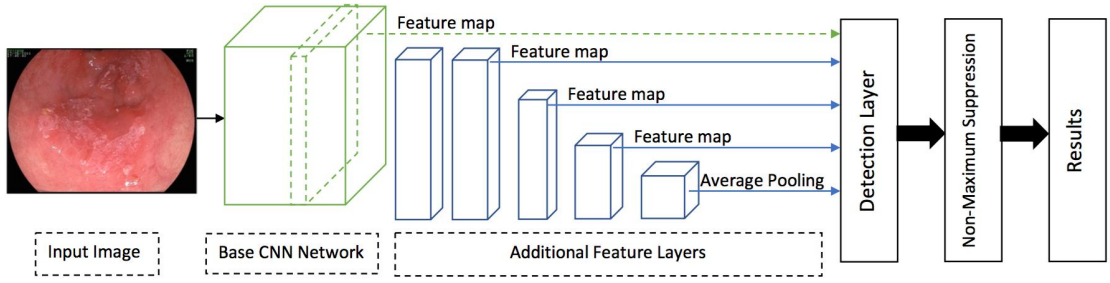


Figure 4.12: General architecture of the **SSD** (Wei Liu et al., 2016). The SSD is a single unified network for both testing and inference.

and choosing the box with the highest confidence. Additionally, negative samples are kept with a ratio of 3:1 compared to positive samples to apply *Hard-Negative Mining*. The hard-negative mining helps the network to better learn the incorrect detection leading to more accurate results.

The backbone CNN network used in the Faster R-CNN and the SSD is the VGG'16 (Simonyan and Zisserman, 2014) after discarding the fully connected layer and using its feature map. One of the main reasons for using the VGG'16 is that it has a very high performance towards image classification problems.

4.5 Methods

After evaluating the performance of the different deep learning methods (as the results will be seen in Section 4.6.4), the Faster R-CNN and the SSD showed to have the highest performance in detecting the different abnormalities. Although the SSD had a faster performance in terms of time the Faster R-CNN proved the ability to have a more localized detection with less false positives. Additionally, using the VGG'16 might fail in detecting small scale objects due to information loss (Cao et al., 2017), therefore it might not be able to successfully detect the small abnormal regions with challenging appearances. In this section, to improve the detection performance we propose two novel models that rely on the Faster R-CNN to detect abnormal regions from endoscopic images.

4.5.1 DenseNet based Faster R-CNN with Gabor Features

In this section, we introduce our proposed esophageal abnormality detection method. The entire proposed model is shown in Fig. 4.13. The first step is to extract features from the input endoscopic images using the suggested DenseNet architecture. Next, the RPN generates proposals for abnormality location using the feature map generated by DenseNet. Afterward, several Gabor filter responses are extracted and concatenated with the CNN features from the DenseNet. The fused features are then used as the input to the ROI pooling layer for the final classification of each proposal generated from the previous RPN stage. The implementation details of each step will be explained in the following subsections.

DenseNet

DenseNets (G. Huang et al., 2017) has been introduced recently in the literature. It reduces the connection between the input and output which helps in overcoming the vanishing gradient problem. Each layer in the DenseNet has a reduced feature map size, which is important for training the CNN's on a small dataset leading to less probability of facing the over-fitting problems and to ensure that there is no loss in the transmitted information (Y. Liu et al., 2018). Additionally, each

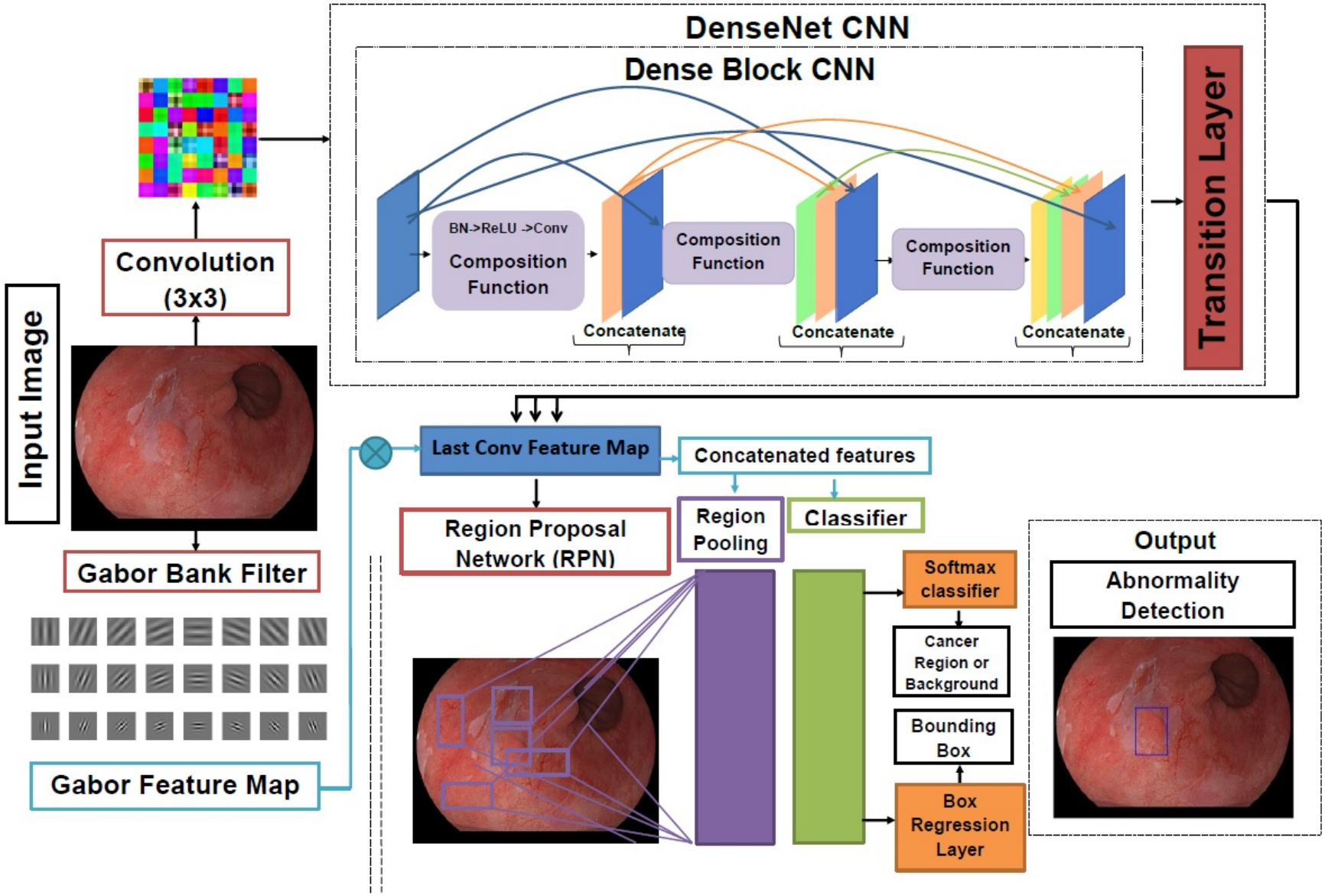


Figure 4.13: The Faster R-CNN framework outline for esophageal abnormality detection in the endoscopic images using DenseNet as a base CNN network and incorporating the Gabor features in the final detection stage. A sample of the densenet architecture with one dense block and a transition layer is illustrated as an example. The denseblock shown demonstrates the connectivity of the concatenated feature map with internal four layers.

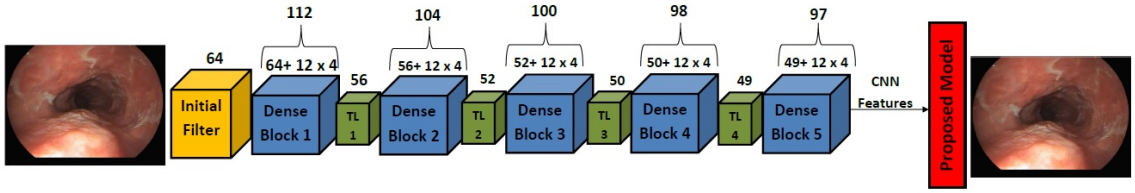


Figure 4.14: General architecture of the proposed DenseNet. An initial convolutional filter of size 64 is first performed on the input image before passing it to the first denseblock. Above each denseblock the feature map size is calculated using the number of internal layers (M) and growth rate (G). A transition layer (TL) exists between each desneblock that changes the size of the feature map.

layer receives supervision from the loss function and a regularizing effect through shorter connections leading to an easier training process. In a tradition feed-forward convolutional network the output of each layer (l) is then connected directly to the input of the next layer ($l+1$) as follows:

$$x_l = H_l(x_{l-1}) \tag{4.5}$$

where H_l represents the operation of the *composite function* that can include any operation such as convolution, pooling, activation function, etc. In literature, the ResNet introduced the concept of skip connection which integrates the output from layer (l) with an identity function to augment the information as follows:

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{4.6}$$

The ResNets allow the gradient to flow directly from later layers to earlier layers through the identity function (i.e. ResNet architecture is shown in Fig. 4.6). However, the process if summation that combines the output of H_l with the identity function may block the flow of information through the network. The DensNet takes the motivation behind the ResNet to another level by suggesting another connectivity concept where all feature maps from the previous layer are available to all upcoming layers. The DenseNet is mainly composed of DenseBlock, Transition Layer and Growth Rate:

- **Dense Block:**

  Each DenseNet is composed of N Dense Blocks. Inside each Dense Block there exists L layers where each layer is connected to all the consecutive layers in a feed forward manner. If x_l is denoted as the output from the l^{th} layer then it is computed as:

  $$x_l = H_l([x_1, x_2, ..., x_{l-1}]) \tag{4.7}$$

  where H_l represents *composite function* in this layer and a concatenation function is processed between each feature layer inside it. The concatenated features are processed through the ***composite function*** (H_l) that consists of Batch Normalization (BN), ReLu and Convolution (3x3). An example of the internal structure of denseblock that is passed on to the Transition layer is shown in Fig. 4.13.

- **Transition Layer:**

  Between each Dense Block, a layer is introduced to decrease the spatial dimension of the features maps called ***transition layer***. It is composed of Conv (1x1) and Average Pooling (2x2).

- **Growth Rate:**

  The output from each concatenation function in eq. (4.7) is feature map f. The size of the L^{th} layers is $f.(l-1)+f_0$, where f_0 is the number of channels of the original input image. In order to improve the parameter efficiency and control the growing of the network, the size of f is limited to a ***growth rate*** (G) with a small integer value. This variable helps regulating the amount of new information each layer holds.

Fig. 4.14 illustrates a general outline of the DenseNet with a description of the feature map size (based on $L = 4$ & $G = 12$) at each block. Additionally, we illustrate samples from the generated feature map using the proposed DenseNet for the endoscopic images in Figs. 4.15- 4.17.

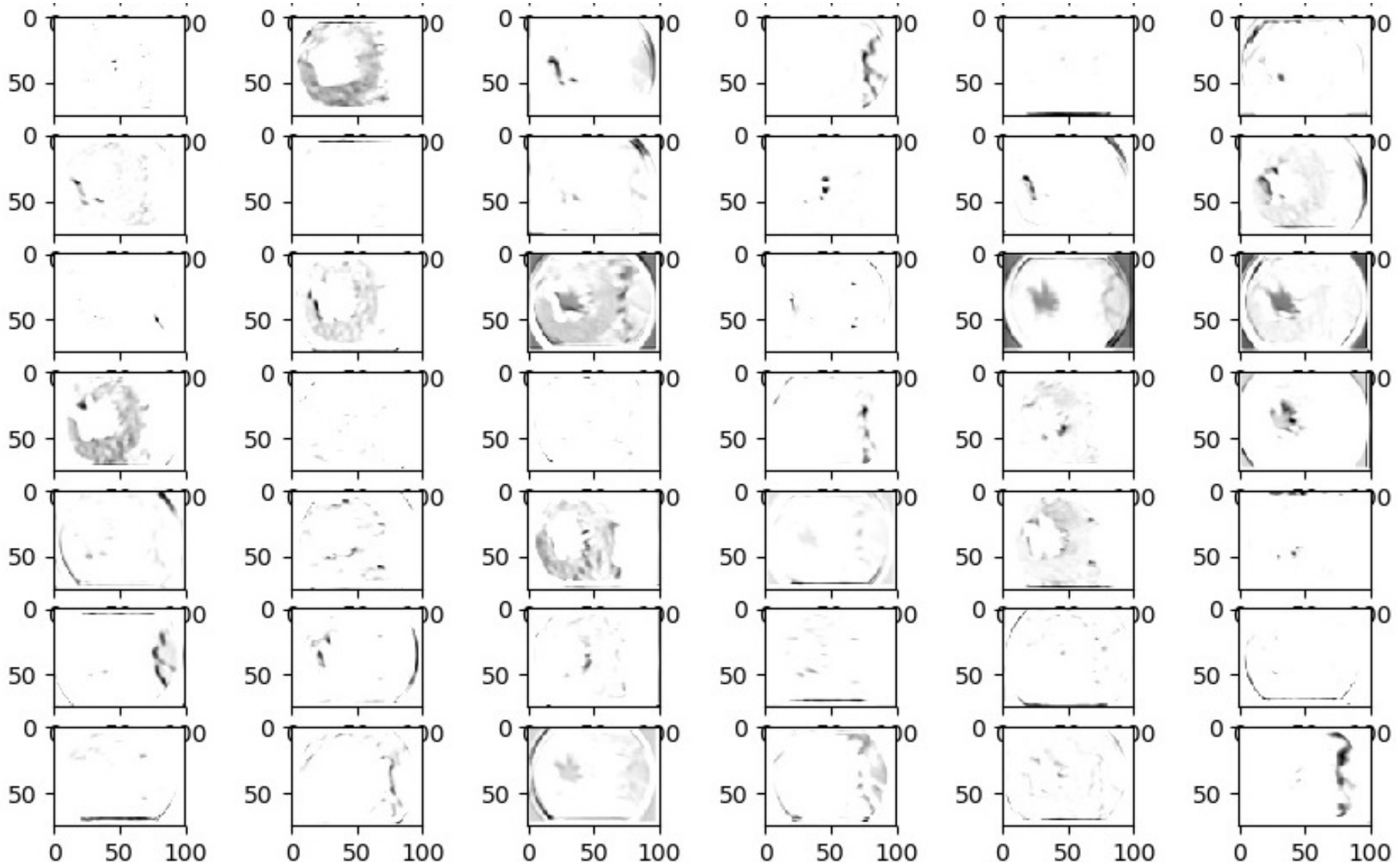


Figure 4.15: Example 1 for different internal feature maps generated by the proposed DenseNet

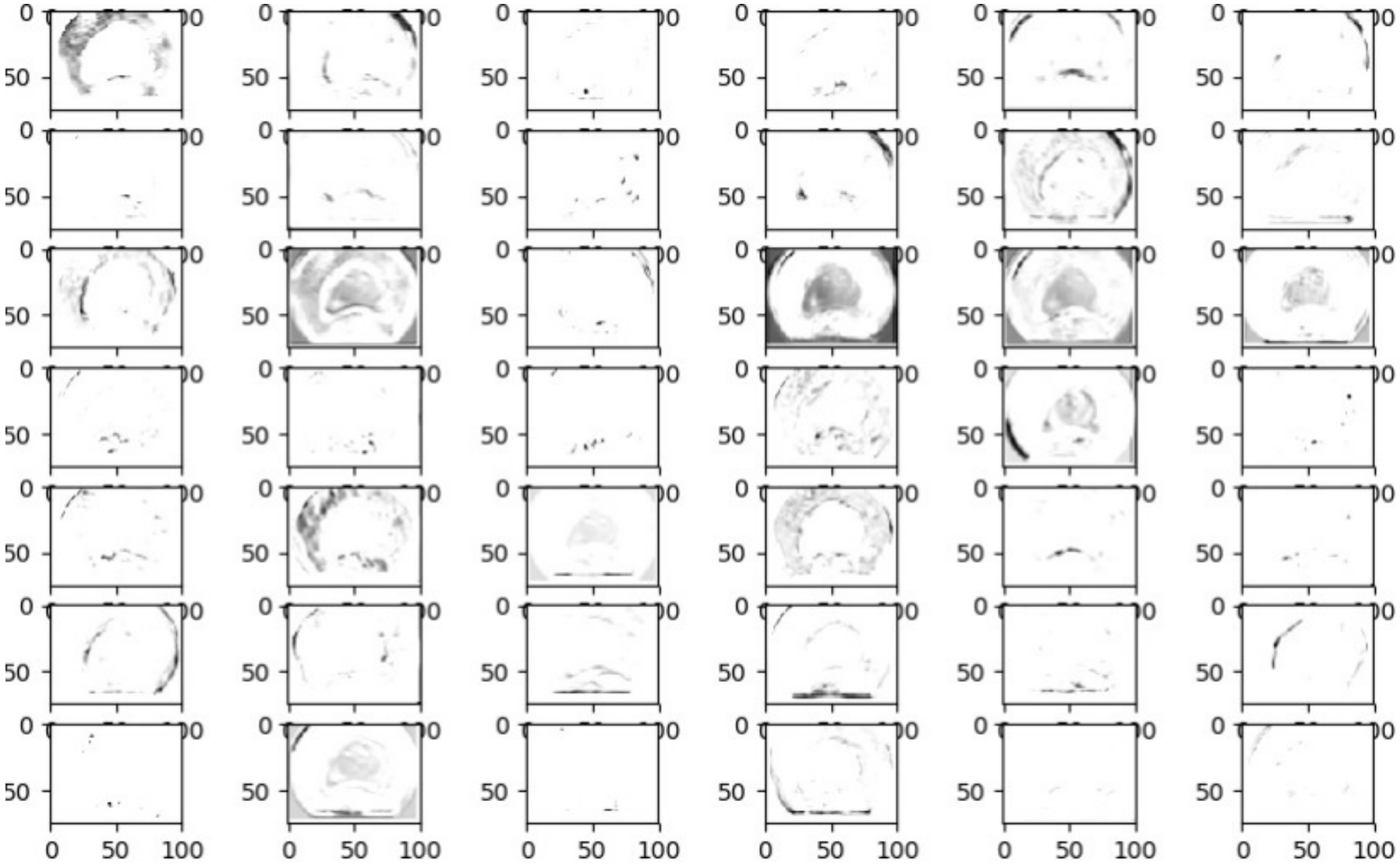


Figure 4.16: Example 2 for different internal feature maps generated by the proposed DenseNet

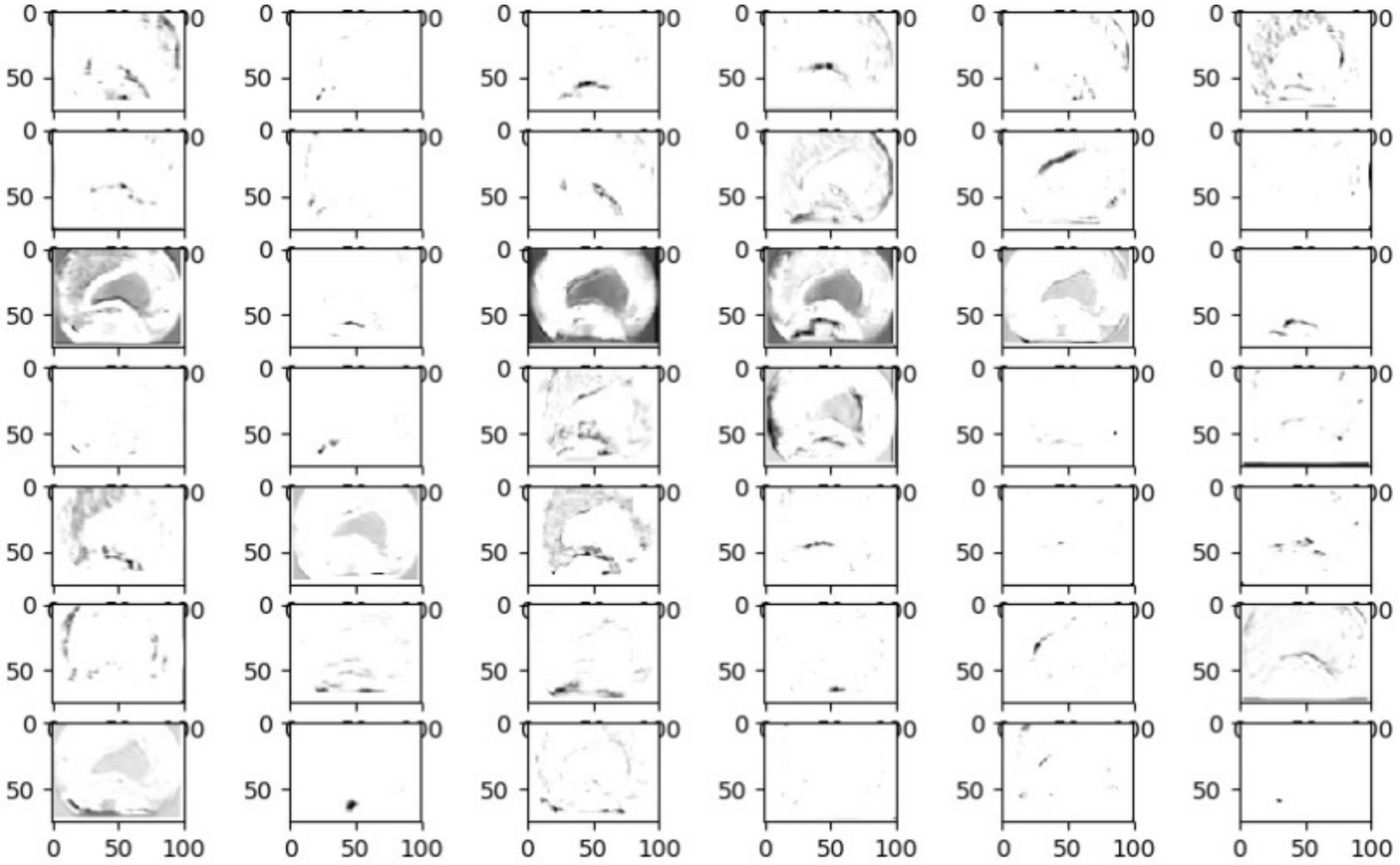


Figure 4.17: Example 3 for different internal feature maps generated by the proposed DenseNet

Gabor Features

The Gabor filter is well known for texture feature representation and has been widely used in pattern analysis applications (Fogel and Sagi, 1989). Gabor filters provide effective textural descriptors by analyzing the local dependencies in both spatial and frequency domains. Generally, a Gabor filter is composed of two parts *(real and imaginary)* representing the orthogonal direction. The Gabor kernel is defined as follows:

$$G(x, y, \theta_k, \lambda) = \exp\left[-\frac{1}{2}\left\{\frac{A_{\theta_k}^2}{\sigma_x^2} + \frac{B_{\theta_k}^2}{\sigma_y^2}\right\}\right] \exp\left\{i\frac{2\pi A}{\lambda}\right\} \tag{4.8}$$

where, λ is the wavelength and i provides the central frequency of the sinusoidal plane wave at an orientation θ_k. The orientation of $\theta_k = \frac{\pi(k-1)}{n}$ where $k = 1, 2, 3.., n$ and n demonstrates the numbers of orientations. The terms A and B are calculated from the spatial orientation of the filter (θ) defined as:

$$A = xcos(\theta_k) + ysin(\theta_k) \tag{4.9}$$

$$B = -xcos(\theta_k) + ysin(\theta_k) \tag{4.10}$$

Finally, σ_x and σ_y denote the standard deviations of the Gaussian envelope along the x and y axes. Fig. 4.18 demonstrates a set of Gabor filters with different sizes, directions, and wavelengths of the sinusoid. The response of the Gabor filter is produced by convolving each filter with the input image by:

$$G_f = I(x, y) \otimes f(x, y, \theta_k, \lambda) \tag{4.11}$$

where, I is the endoscopic input image and $\otimes$ symbolize the convolution operation with the filters generated in different orientations and scales defined in eq.(4.8). Fig. 4.19 shows an example of the Gabor filter responses to endoscopic images from our dataset with 16 different orientations (θ).

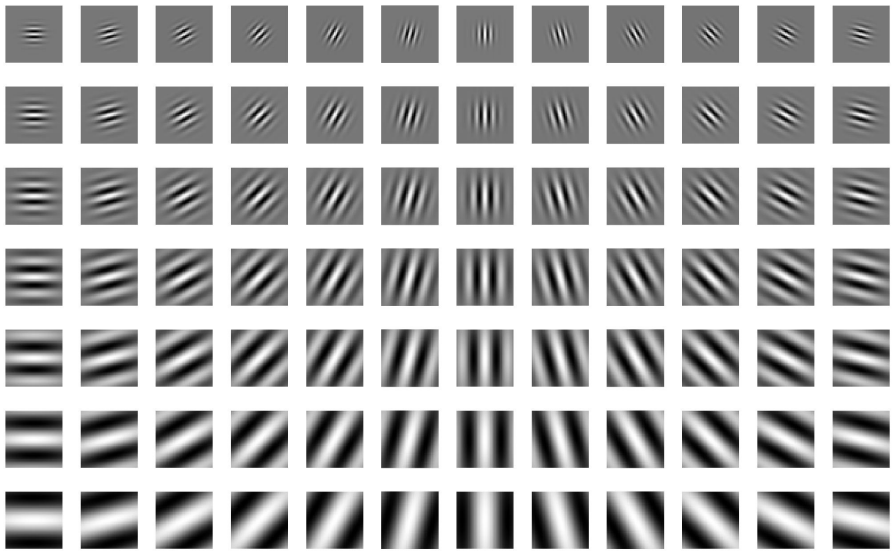


Figure 4.18: Set of Gabor filters with different sizes, directions, and sinusoid wavelengths.

Feature Map Concatenation Fusion

As explained earlier, to produce the output bounding box prediction, the ROI-pooling is performed on the feature map layer generated by the CNN network. In the proposed model, a Gabor feature map is generated by convolving the endoscopic image with a set of Gabor filters with different orientations. This Gabor feature map is combined with the final DenseNet feature map using concatenation fusion (Feichtenhofer, Pinz and Zisserman, 2016), the fused features are then used by the ROI pooling stage. The concatenation fusion takes place as:

$$F_{map} = concatenate(f_{dense}, f_{gabor}) \tag{4.12}$$

where, the two feature maps are stacked at the same spatial location of (i, j). Therefore, more detailed information is provided to the bounding box detection and classification from the newly concatenated feature map.

The proposed detection model showed an improved performance in detecting different abnormalities from the endoscopic images. However, the process of training and test-

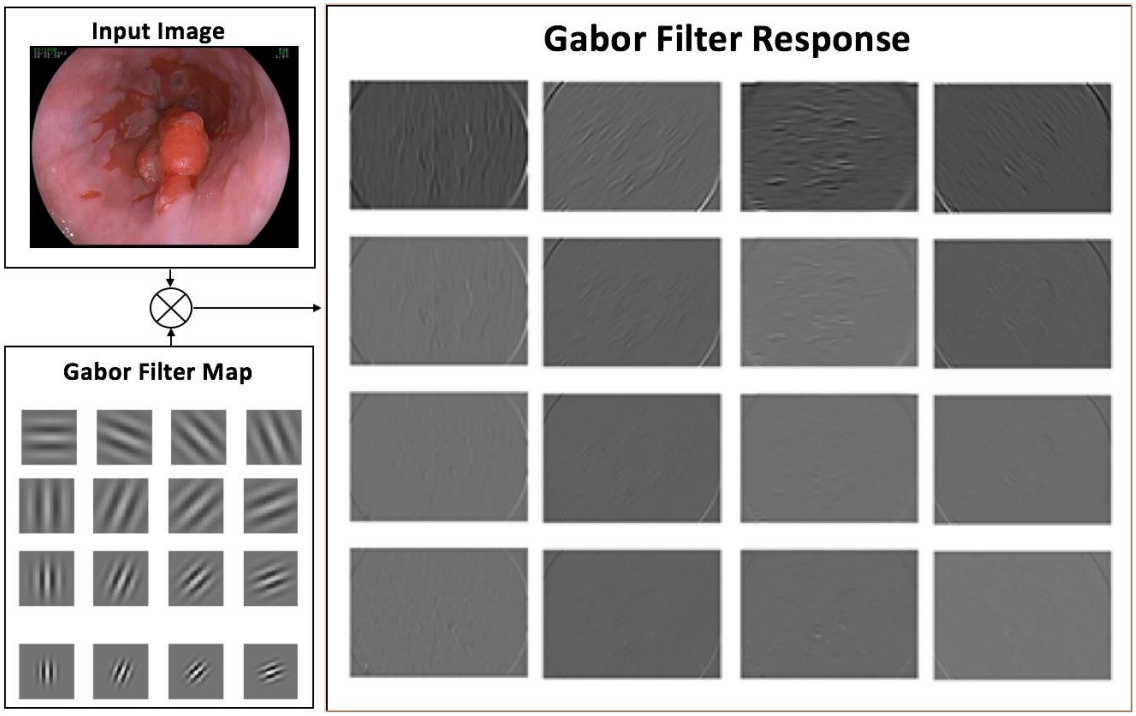


Figure 4.19: An example of the Gabor Filter response from the HD-WLE endoscopic image, obtained by convolving the image with the Gabor kernels in the filter bank with kernel size =5 with 16 different orientations.

ing the network with handcrafted features had a very slow performance. Moreover, after investigating the results, we found that the performance of the model can be improved by benefiting from the Gabor filter (i.e. which proved its efficiency) in a deep learning network. Therefore, in the next section, we proposed a new two input network method that includes Gabor feature in a different way leading to an elevated performance.

4.5.2 GFD Faster R-CNN

A novel GFD Faster R-CNN model is proposed to automatically detect esophageal abnormalities. The main framework of the model is presented in Fig. 4.20. As shown, first, we generate a Gabor Fractal (GF) image from the original endoscopic image which is later used as a second input in our model. Then we introduce the DenseNet to learn features from both images independently. The CNN features of the endoscopic image are used by RPN stage to obtain candidate region proposals. Later, the features from both images are combined together through bilinear fusion

presenting a pairwise interaction between the two feature maps, so providing informative feature representation. Finally, the fused features are used in the ROI pooling stage for the final abnormality detection.

Two-input Faster R-CNN

The baseline of the proposed model is the Faster R-CNN (Ren et al., 2015). It is composed of two stages: *Region Proposal Network (RPN)* and *Region-of-Interest (ROI) pooling layer* (i.e. as mentioned earlier in Sec. 4.4). As a recall, the RPN is responsible to generate a list of region proposals that might be an abnormality. The RPN relies on *anchor boxes* to produce K proposals for each location (as shown in Fig. 4.20 (*blue dotted box*)). For each image, there exist ($W \times H \times K$) proposals where W and H represent the size of the feature map. The input of the ROI pooling is dependent on the output from the RPN layer. The ROI pooling unifies the size of the feature map for each proposal and classifies them using softmax into abnormal or normal, while the regression layer is used to give the coordinates of the output bounding box (c_x, c_y, w, h). In our model, the RPN only uses the CNN feature maps of the original endoscopic image to generate candidate region proposals. Features from the original and GF images are fused using bilinear fusion before the ROI pooling stage for final detection output (based on the proposals generated by the RPN). The total loss function of our proposed model is defined as:

$$L_{total} = L_{rpn} + L_{fusion}(f_{rgb}, f_{gf}) \tag{4.13}$$

where L_{total} represents the total loss of the model, the L_{rpn} denotes the loss of the RPN network and L_{fusion} the loss of the ROI classifier from the fused features map (f_{rgb}: original endoscopic image features, f_{gf}: GF image features). Both L_{rpn} and L_{fusion} have two loss terms: the classification accuracy and the regression loss of the bound box coordinates of the predicted output. The loss functions for each stage were measured as the default setting for the Faster R-CNN as described in (Ren et al., 2015).

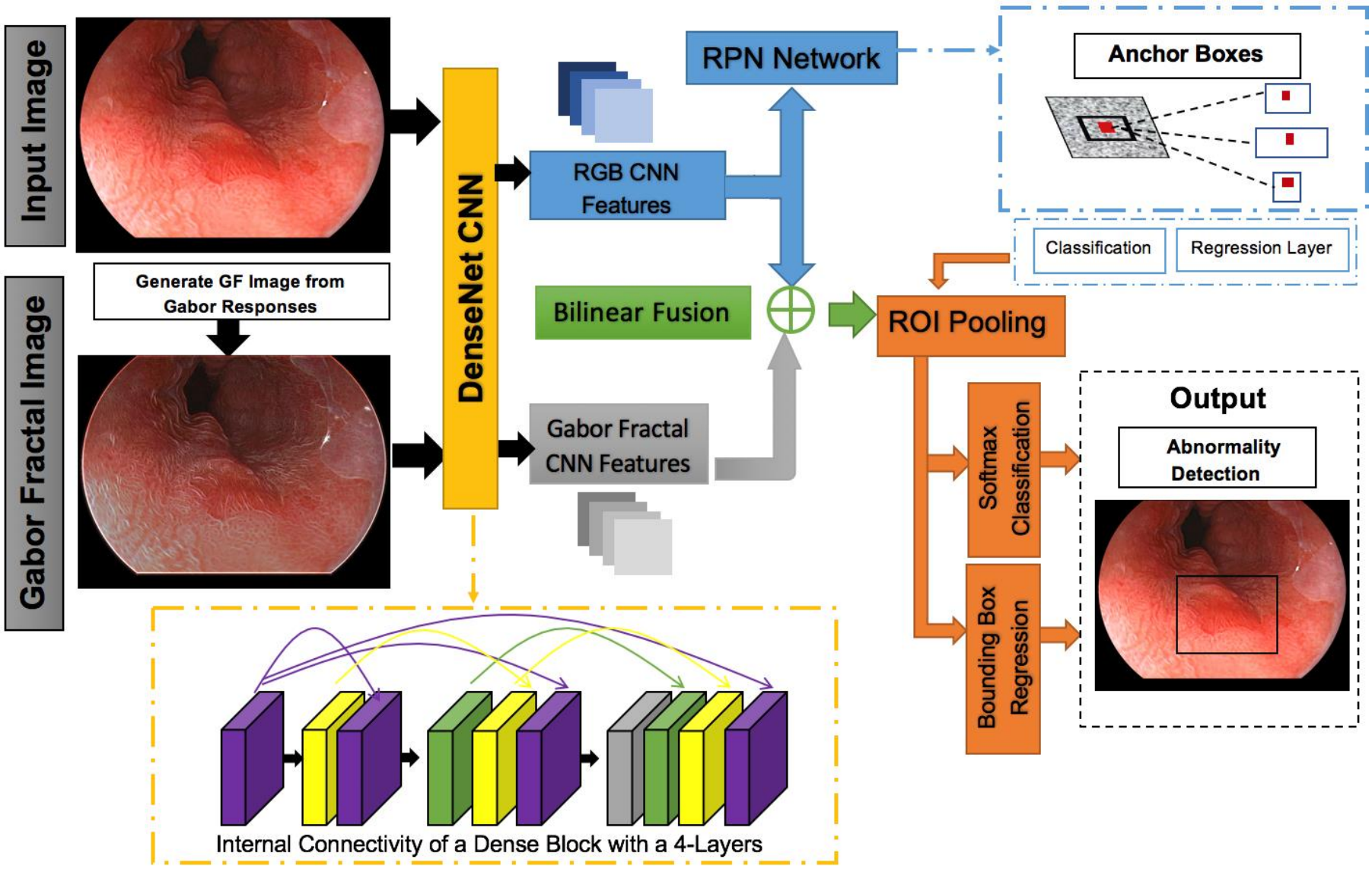


Figure 4.20: The proposed GFD Faster R-CNN framework. The GF image is first produced by extracting different Gabor filter responses from the endoscopic image. Proposals are generated through the RPN stage using anchor boxes and CNN features of the endoscopic image only. Features from the two images are fused using bilinear fusion before ROI pooling stage for final detection of abnormality location. The DenseNet is used as a backbone CNN to learn features.

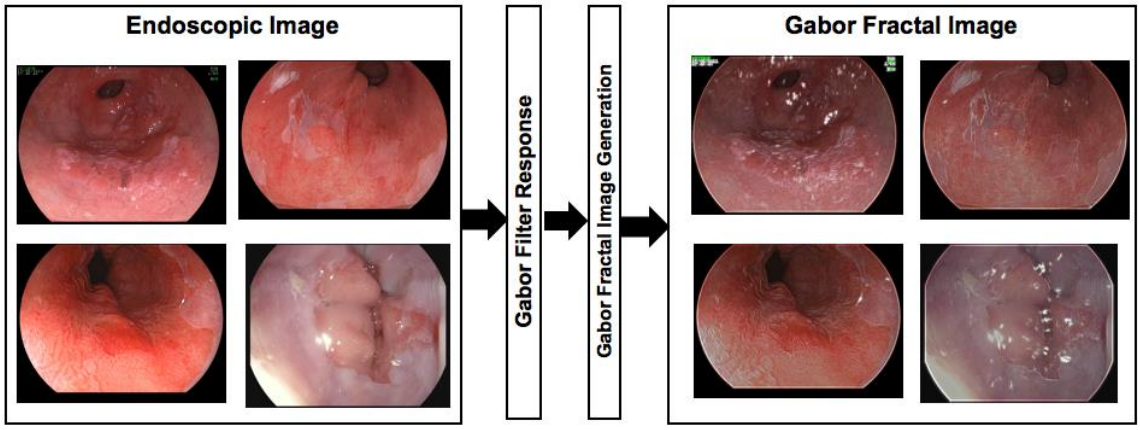


Figure 4.21: Examples of the generated GF images. The Gabor filter response are extracted from different orientations and scales to generate the GF image.

Gabor Fractal

In our model, we propose a Gabor Fractal (GF) image by generating different Gabor filter responses (using eq. 4.11) from the original endoscopic input image and merging them. The GF image is used as a second input in our model as shown in Fig. 4.20. The use of Gabor features has shown a remarkable effect in detection methods (Yao et al., 2016; Kwolek, 2005) and proved its ability to improve the representation of deep features (Luan et al., 2018). The steps for generating the GF image is in Algorithm 1. First, we initialize the GF_{img} equal to the first filter response G_{f1}. Then we keep looping on all N filter responses and compare the current pixel value in GF_{img} with the one in G_{fi}. The maximum value is selected as the new value in GF_{img}. The process of merging the filter responses using all the generated filters G_f are used to produce the GF image as follows:

$$GF_{img}(x, y) = Max(\forall\, G_{f_i}(x, y)) \qquad \{i = 1, 2, ..., N\} \tag{4.14}$$

where, Max is the maximum pixel value at each location (x, y) for all the generated (N) number of Gabor filter responses (eq. 4.11). Fig. 4.21 demonstrates different examples of the endoscopic images and their corresponding generated GF images. As shown, the GF image emphasizes the hidden fractal features in the image which complements the feature representation extracted through CNN.

Algorithm 1 Generation Gabor Fractal Image (GF_{img})

Require: Gabor Filter responses (G_{f_i})
Ensure: GF_{img}
Initialize GF_{img} equal to first Gabor Filter ($G_{f_{i=1}}$)
for $i = 2$ to N **do**
  for $x < W$ **do**
    for $y < H$ **do**
      if $GF_{img}(x, y) < G_{f_i}(x, y)$ **then**
        $GF_{img}(x, y) \Leftarrow G_{f_i}(x, y)$
      end if
    end for
  end for
end for

Feature Map Fusion

In the last stage before the ROI pooling, the CNN features produced from both the original endoscope and Gabor Fractal images are combined through *Bilinear Fusion* to improve the final detection performance. The *Bilinear Fusion* (Feichtenhofer, Pinz and Zisserman, 2016) computes a matrix from the outer product of each location from both feature maps followed by global average pooling as defined in the following equation:

$$F_{bil} = \sum_{i=1}^{H} \sum_{j=1}^{W} F_{i,j}^{rgbT} \odot F_{i,j}^{gf} \tag{4.15}$$

where, F^{rgb} is the feature map from the original endoscopic image, F^{gf} is the feature map from gabor fractal image, T is transpose, (H,W) represent the height and width of the feature map and (i, j) represent the location within feature map.

4.6 Experimental Setting and Results

This section will describe the evaluation results of the proposed automatic esophageal abnormality detection methods from still images. First, the datasets will be mentioned, followed by the experimental setting and evaluation methods. Thereafter, the results will be discussed and the statistical analysis will be explained. Moreover,

we demonstrate different visual examples of the detection output from the utilized dataset using the proposed models.

4.6.1 Dataset

For all of the experiments, the systems are trained and tested using two datasets separately: the Kvasir and the MICCAI'15. The dataset is composed of 1000 images for Kvasir dataset and 100 images gathered from 39 patients (as mention in Chapter 2, Sections 2.5.3 and 2.5.2). The Kvasir includes only the esophagitis abnormality while the MICCAI'15 includes EAC regions. Since deep learning requires a large amount of data, *Data Augmentation* is introduced to the training data to increase the dataset to achieve better performance. It contains random rotation in different directions (45°, 135°, 225°), flipping, stretching vertically and horizontally for only 30% of the training dataset selected randomly. Therefore, the Kvasir dataset after augmentation is increased to 1900 images while the MICCAI'15 dataset reaches 280 images. The augmented images are only included in the training phase.

4.6.2 Implementation Setup

In the RPN layer of the Faster-RCNN network, we adjust the anchor box numbers and sizes to the default setting as proposed in (Ren et al., 2015). There exists k=9 anchors at each location with 3 scales (128^2, 256^2, and 512^2 pixels) and 3 aspect ratios (1:1, 1:2, and 2:1). Additionally, the loss function of the RPN stage during training process is defined as:

$$L(\hat{p}_i, \hat{t}_i) = \frac{1}{N_c} \sum_i L_c(\hat{p}_i, \breve{p}_i) + \lambda \frac{1}{N_r} \sum_i \breve{p}_i L_r(\hat{t}_i, \breve{t}_i) \tag{4.16}$$

where, the index of an anchor is denoted by i, $\hat{p}_i$ and $\breve{p}_i$ respectively, representing the prediction and the ground-truth of the anchor i, being an abnormal region in the image or not. In the same manner, $\hat{t}_i$ and $\breve{t}_i$ denote the coordinates of the predicted bounding box by RPN and the ground-truth one. The total number of inputs are represented by N_c for classification layer and N_r for regression layer that is weighted

by a balancing parameter λ. The L_c defines the classification loss by taking the log loss function over two classes (*abnormal candidate or not*) defined as:

$$L_c(\hat{p}_i, \breve{p}_i) = -\breve{p}_i \log \hat{p}_i - (1 - \breve{p}_i) \log(1 - \hat{p}_i) \tag{4.17}$$

And, L_r represents the regression loss defined as:

$$L_r(\hat{t}_i, \breve{t}_i) = L_1^{smooth}(\hat{t}_i - \breve{t}_i) \tag{4.18}$$

The regression loss (L_r) is only active if the ($\hat{p} = 1$) which means that the anchor boxes returned a positive candidate and it is deactivated if ($\hat{p} = 0$).

To select the parameters of building the DenseNet, different values for the *dense blocks*, *no. of layers* (L) and *growth rate* (G) were assessed on the dataset. The optimal DenseNet network performance in our model is formed of 5 *dense blocks* with $L = 4$ and $G = (12, 16)$. Furthermore, the transition layer applied between each *dense block* is made of (1x1) convolution layer and (2x2) average pooling layer. An initial filter of size 64 is applied to the endoscopic input image using a (3x3) convolution to create a feature map for the first denseblock (as shown in Fig. 4.14).

For the orientation of the Gabor filters, sixteen degrees with $\theta = (0, \frac{\pi}{16}, \frac{\pi}{8}, \frac{3\pi}{16}, \frac{\pi}{4}, \frac{5\pi}{16}, \frac{3\pi}{8}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{5\pi}{8}, \frac{11\pi}{16}, \frac{3\pi}{4}, \frac{13\pi}{16}, \frac{7\pi}{8}, \frac{15\pi}{16}, \pi)$ chosen to ensure covering the whole space of the region with a reasonable step. The maximum and minimum values for the Gabor filter size were selected empirically by visually inspecting the filter response to the input image and the kernel sizes is set to $k = 5$.

The weights are initialized randomly with a gaussian distribution ($\mu = 0$, $\sigma = 0.01$). The initial learning rate was set to 0.0003 and drops by the factor 0.1 every 1000 iteration and used a weight decay of 0.0004. The model is implemented using Keras Libary (Tensorflow backend) on a desktop with Intel Core i7 (3.6GHz processor) and an NVIDIA GeForce GTX1080 Ti with 11GB on a single GPU memory.

4.6.3 Evaluation Measures

For the Kvasir and MICCAI'15 datasets, the process of automatically detecting the abnormal regions is evaluated (i.e. precancerous and cancerous regions) using the standard measures *Recall, Precision, Specificity* and *F-Measure* (as explained in Chapter 2 in Section 2.6) to compare with the ground truth annotation. The IoU is used to measure the overlap ratio between the detection results and the manual segmented gold standard which was explained in Chapter 2 (Section 2.6 and Eq. (2.6)).

4.6.4 Experimental Results and Discussion

To evaluate the performance of the proposed methods, three sets of comparative experiments were investigated on the two available datasets as follows:

Evaluation of Deep Learning Methods Results

The four deep learning object detection approaches discussed in section 4.4 have been carried on the two datasets. In this experiment, if the IoU value between the generated bounding box and the ground truth is less than 0.5 then the produced bounding box is considered to be a false prediction (non-cancerous). Furthermore, the time for the detection processes for each method was measured in seconds during the testing phase.

The experiments have been carried out using three types of validation for the MICCAI'15 dataset. ***Experiment 1:*** from the 39 patients, 50% were used for training (21 patients *(12 cancerous, 9 non-cancerous barrett's)*), 25% for validation (9 patients *(5 cancerous, 4 non-cancerous barrett's)*) and 25% for testing (9 patients *(5 cancerous, 4 non-cancerous barrett's)*). The experiments were carried twice to verify the results using more cases by changing the patients dataset between the validation and testing sets in the second experiment. Therefore, the results presented in Table 4.1 are based on a total of 18 patients (10 cancerous and 8 non-cancerous barrett's) that are entirely different from the dataset used for training the model. ***Experiment 2:*** The dataset was evaluated based on 5-fold-cross-validation (5-fold-CV),

where the dataset is divided into 5 folds randomly (Each fold will hold 7∼8 patients). The results of the second experiment are shown in Table 4.2. ***Experiment 3:*** Leave-One-Patient-Out cross-validation (LOPO-CV) is applied to compare the four detection methods. Table 4.3 demonstrates the results from LOPO-CV experiment in addition to a comparison with two state-of-the- art (Mendel et al. (Mendel et al., 2017) and Sommen et al. (Sommen et al., 2016)) methods that use the same dataset. The results of the three experiments will be discussed further in the following section.

Table 4.1: Sensitivity (SE) and Specificity (SP) and F-Measure (FM) for the state-of-the-art object detection deep learning methods on the MICCAI'15 dataset based on 50% training, 25% validation and 25% testing.

Method	**SE** (%)	**SP** (%)	**FM** (%)
R-CNN	47.0	41.0	44.0
Fast R-CNN	53.0	57.0	55.0
Faster R-CNN	72.0	83.0	83.0
SSD	**93.0**	**93.0**	**93.0**

Table 4.2: Sensitivity (SE) and Specificity (SP) and F-Measure (FM) for the state-of-the-art object detection deep learning methods on the MICCAI'15 dataset based on **5-fold-CV**.

Method	**SE** (%)	**SP** (%)	**FM** (%)
R-CNN	50.0	40.0	48.0
Fast R-CNN	64.0	64.0	64.0
Faster R-CNN	78.0	80.0	79.0
SSD	**90.0**	**88.0**	**88.0**

Table 4.3: Sensitivity (SE) and Specificity (SP) and F-Measure (FM) for the state-of-the-art object detection deep learning methods on the MICCAI'15 dataset based on **LOPO-CV**.

Method	**SE** (%)	**SP** (%)	**FM** (%)
R-CNN	60.0	56.0	59.0
Fast R-CNN	64.0	60.0	63.0
Faster R-CNN	88.0	86.0	87.0
SSD	**96.0**	**92.0**	**94.0**
Mendel *et al.* (Mendel et al., 2017)	94.0	88.0	91.0
Sommen *et al.* (Sommen et al., 2016)	86.0	87.0	87.0

Furthermore, the bounding box results from each method have been provided on some sample images shown in Fig. 4.22 and compared to the ground-truth bounding box. The figure shows different samples of true and false positive detections. An example from one non-cancerous image that had a false prediction by the R-CNN and Fast R-CNN method is shown in Fig. 4.22f and another one by the R-CNN is shown in Fig. 4.22l. Moreover, Fig. 4.22j illustrates the detection of Faster R-CNN and SSD only as the other two methods failed to find an EAC region. The rest of the figures demonstrate the performance of the four models in detecting the abnormal regions in minor and complex tumors.

Table 4.4: Sensitivity (SE) and Precision (Pre) and F-Measure (FM) for the state-of-the-art object detection deep learning methods on the Kvasir dataset based on 50% training, 10% validation and 40% testing.

Method	**SE** (%)	**Pre** (%)	**FM** (%)
R-CNN	64.3	69.8	66.9
Fast R-CNN	70.7	73.5	72.1
Faster R-CNN	**83.6**	**86.1**	**84.8**
SSD	80.1	78.4	79.2

For the Kvasir dataset, since there is no information about the number of patients or images per patient were provided, only one experiment is carried out where the dataset was divided randomly 50% training, 10% validation, and 40% testing. The results of this experiment are shown in Table 4.4. Moreover, in Fig. 4.23 we represent the bounding-box output from the four models and compare them to the ground truth by the expert. The figure shows samples of the true and false positives by each method. For example, a true positive detection is given by the four different methods in Fig. 4.23f. While in another image a true positive is given by Faster R-CNN and SSD and false-positives are given by R-CNN and Fast R-CNN in Fig. 4.23l. The remaining figures show different examples of the output that will be discussed later.

The sensitivity, specificity, and F-measure are measured for the three experimental validation methods on the MICCAI'15 dataset. Results in Table 4.1 are based only on 18 patients (10 cancerous and 8 non-cancerous barrett's) as described at the beginning of this section. The SSD outperforms among the compared methods with a result of 93% for the SE, SP, and F-measure. The high sensitivity of the SSD result

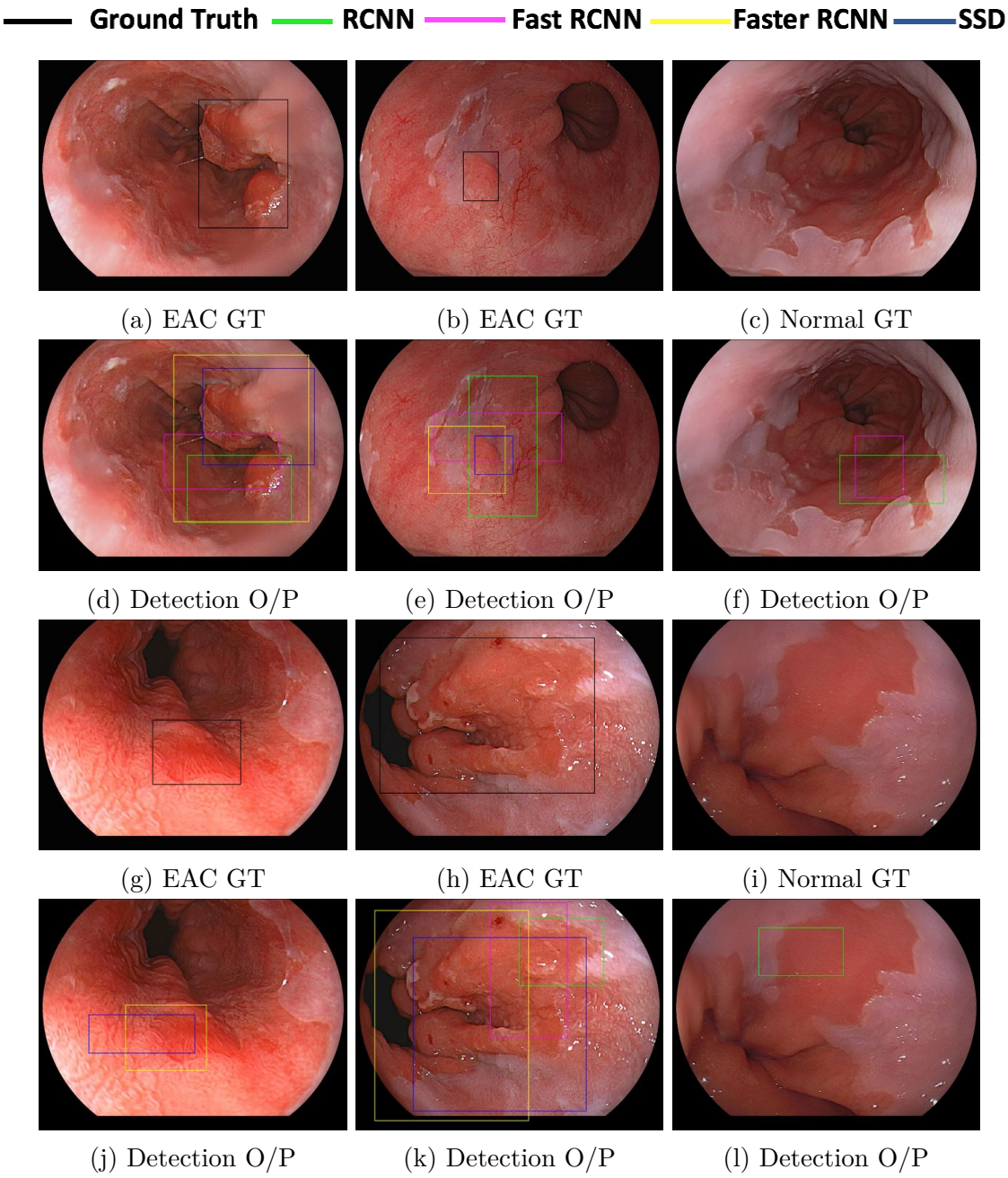
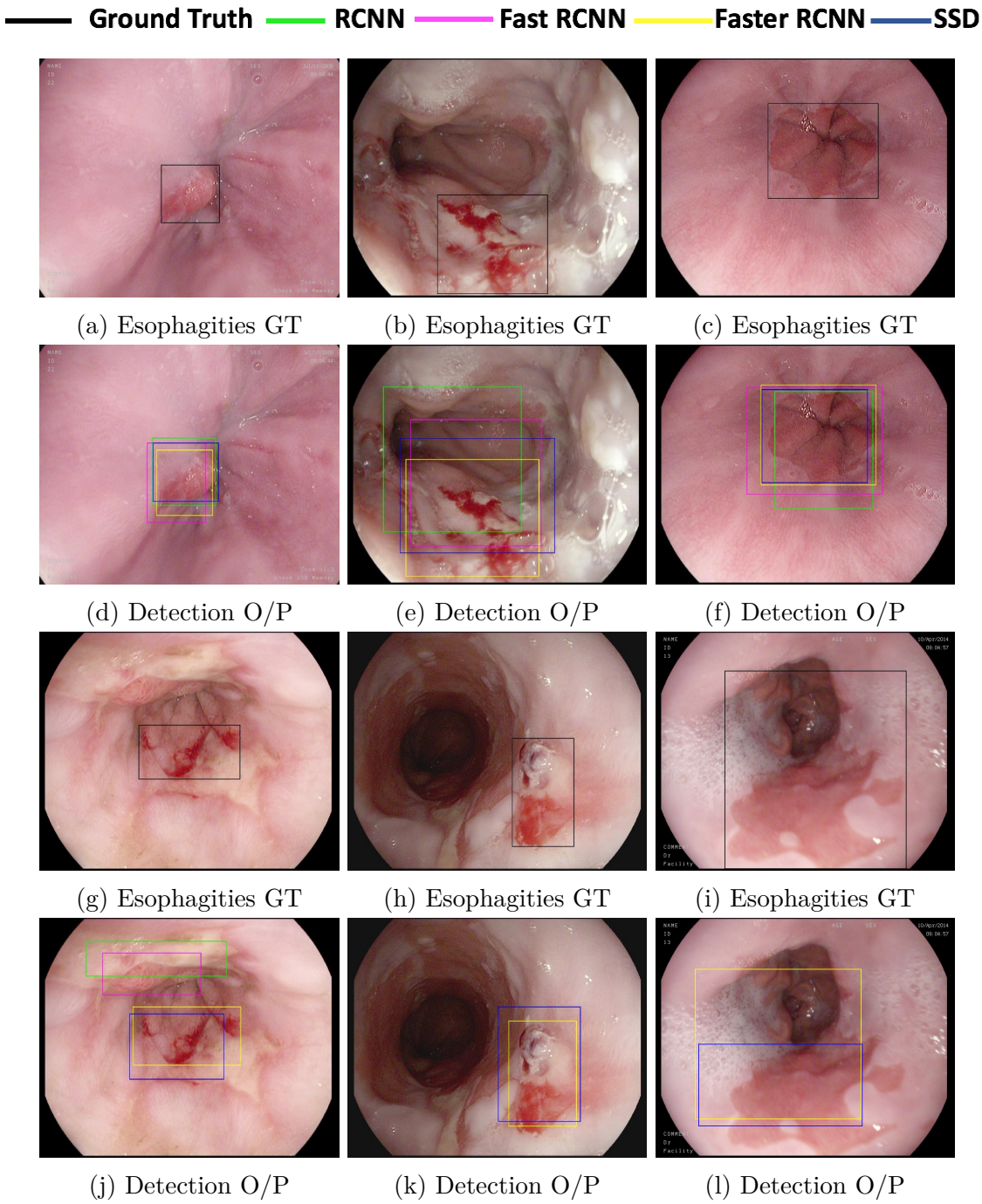


Figure 4.22: Bounding-box ground truth based on experts annotation and the output from the R-CNN, Fast R-CNN, Faster R-CNN and SSD when using 5-fold-CV from different patients using Miccai'15 dataset. Showing correct prediction in (d, e, j & k) with different scores and a false prediction on a non-cancerous patient in (f &l).



(a) Esophagities GT (b) Esophagities GT (c) Esophagities GT

(d) Detection O/P (e) Detection O/P (f) Detection O/P

(g) Esophagities GT (h) Esophagities GT (i) Esophagities GT

(j) Detection O/P (k) Detection O/P (l) Detection O/P

Figure 4.23: Bounding-box ground truth based on expert annotation and the output from the R-CNN, Fast R-CNN, Faster R-CNN, and SSD when using 5-fold-CV from different patients using Kvasir dataset. Showing correct prediction in (d, f, k & l) with different scores and a false prediction two methods with small IoU in (e & j)

shown in this table indicates that it had a good performance in detecting EAC regions from the cancerous images and less false positives in the non-cancerous barrett's images. The Faster R-CNN followed by with results of 72% for the sensitivity and 83% for both the specificity and F-measure.

From Table 4.2 based on 5-fold-CV. The SSD surpasses the other three methods with a sensitivity of 90% and 88% for both specificity and F-measure. The results demonstrate that the SSD has a high performance in generating bounding boxes that locate abnormal regions throughout the testing dataset and less false ones. For the Faster R-CNN as shown in Table 4.2, the results of the sensitivity were 78% and 80% for the specificity, and 0.79 for the F-measure demonstrating an acceptable performance coming in second place.

As a further study, a comparison of the results with other state-of-the-art models presented by Mendel *et al.* (Mendel et al., 2017) and Sommen *et al.* (Sommen et al., 2016) is illustrated in Table 4.3. For a fair evaluation, we employ the same validation method **LOPO-CV**. Firstly, the sensitivity was evaluated, and the SSD achieved the highest performance among the four deep learning methods and surpassed the results of (Mendel et al., 2017) by 2% and (Sommen et al., 2016) by 10%. Also, the Faster R-CNN outperformed against (Sommen et al., 2016) by 2%. Additionally, the specificity of the SSD achieved 92% indicating the improvement of less false positives regions. While, the Faster R-CNN achieved 86% that is considered comparable with results of (Mendel et al., 2017) and (Sommen et al., 2016).

As observed in Tables 4.2 and 4.3, the R-CNN and the Fast R-CNN have the lowest performance. The reason behind this is that both methods rely on a selective search algorithm to generate regions of interest. As explained in the earlier section, a selective search algorithm uses the greedy algorithm to search for a location for object localization. The greedy algorithm has limitations in finding the optimal solution. Additionally, the grouping process is done based on the color space difference and similarity metrics. While for our dataset, it is difficult to differentiate between non-cancerous barrett's regions and EAC solely based on color as they both have a darker color than normal regions which might lead to more false positives. On

the other hand, the use of anchor boxes and priors of the Faster R-CNN and the SSD help improve the performance of generating more candidate regions of interest. Furthermore, the results of Table 4.3, in general, are more improved than that in Table 4.2 as the LOPO-CV allows more datasets to be trained than the 5-fold-CV.

In Table 4.4, the results for the Kvasir dataset are presented. As shown, the R-CNN and the Fast R-CNN had the lowest performance among the four methods. However, the Faster R-CNN outperformed in detecting the Esophgities region with a sensitivity of 83.6%, a precision of 86.1% and F-measure 84.8%. The detection of precancerous regions (i.e. Esophgities) is more difficult to recognize than cancerous regions as they have properties that might look similar to normal regions. The results demonstrate the efficiency of Faster R-CNN in dealing with challenging properties and locating them.

Moreover, the differences in sensitivity and specificity between the four object detection methods were statistically evaluated using the paired T-test at a confidence level of 95%. The results of the two-tailed p-value of the two best performers (SSD & Faster R-CNN), when compared with the other two methods, are illustrated in Table 4.5. As shown, the difference between the sensitivity and specificity of the SSD and Faster R-CNN was found to be significantly different when they were compared to the R-CNN and Fast R-CNN using the T-test.

Additionally, the T-test was also employed to determine if there are any statistical differences in the sensitivity and specificity, obtained using the two validation methods (i.e. 5-fold-CV (Table 4.3) and LOPO-CV (Table 4.3)). As shown Table 4.6, the p-value of the sensitivity and specificity for each deep learning object detection method was as follows R-CNN (*0.0235,0.0068*), Fast R-CNN (*0.3222, 0.1594*), Faster R-CNN (*0.0238 ,0.0832*) and SSD (*0.0832, 0.1594*). Our analysis based on these p-values suggests that the two validations for the R-CNN and Faster R-CNN shows a significant difference. On the other hand, the difference in results for the SSD and the Fast R-CNN is not statistically significant. The reason behind this is due to the limited dataset used in the current evaluation with only 39 patients.

Moreover, the detection time during testing was measured in seconds for each method

Table 4.5: The *p-value* calculate using the *paired T-test* to measure the difference of sensitivity and specificity results between the four deep learning methods for MICCAI'15.

	Sensitivity		**Specificity**	
Method	R-CNN	Fast R-CNN	R-CNN	Fast R-CNN
Faster R-CNN	0.0049	0.1279	0.0001	0.0443
SSD	0.0012	0.0882	0.0001	0.0036

Table 4.6: The *p-value* calculate using the *paired T-test* to measure the difference of sensitivity and specificity values between the results of 5-fold-CV (Table 4.3) and LOPO-CV (Table 4.3) for the four methods on the MICCAI'15 dataset.

Method	**Sensitivity**	**Specificity**
R-CNN	0.0235	0.0068
Fast R-CNN	0.3222	0.1594
Faster R-CNN	0.0238	0.0832
SSD	0.0832	0.1594

as shown in Table 4.7. The time started with a range of 13.38 $\sim$ 37.81 seconds when using the R-CNN and then decreased while using a more updated method. The R-CNN requires a significant amount of time as it generates around 2000 region proposals for each location and then used to extract features from them using CNN. This leads to a repetition of almost 2000 times to extract features from one image. The detection time drops to 0.65$\sim$ 2.1 seconds when using the Fast R-CNN, as the selective search is applied to the extracted features after applying the CNN to the input image. The Faster R-CNN was faster after sharing the weights and feature map between the RPN and ROI pooling layer resulting in a range of 0.3 $\sim$ 0.4 seconds to generate detection bounding boxes. The SSD surpassed against the other methods in predicting region in most of the cancerous images with only 0.1 $\sim$ 0.3 seconds. The reason for this is that the SSD can localize the object and classify it in a single forward pass network. We believe that with more powerful hardware (i.e. Nvidia Titan, Nvidia Tesla V100), the detection speed would be further increased for any of these methods.

In addition to providing the quantitative evaluation, we also randomly choose some qualitative results of the deep learning object detection methods for different cases as

Table 4.7: Time in seconds (*sec*) for each detection method to generate bounding-box for the abnormal region for both datasets.

	R-CNN	**Fast R-CNN**	**Faster R-CNN**	**SSD**
Time (sec)	13.38 $\sim$ 37.81	0.65 $\sim$ 2.1	0.3 $\sim$ 0.45	**0.1 $\sim$ 0.2**

shown in Fig. 4.22 and Fig. 4.23. Concerning the MICCAI'15 dataset; for example, Fig. 4.22e demonstrates that the different methods can detect some difficult instances in which the abnormality is located in a small region and is visually similar to other areas inside the same image. Also, cases such as Fig. 4.22d and Fig. 4.22k where the abnormal areas are present in most of the images. The SSD and Faster R-CNN show the ability to detect most of the EAC area compared to the ground-truth. Furthermore, Fig. 4.22f and 4.22l list some false positive regions detected by the R-CNN and Fast R-CNN. The non-cancerous barrett's from normal patients have a difference in color in some areas as shown in Fig. 4.22c and 4.22i which makes the detection challenging.

For the Kvasir dataset; Fig. 4.23d and 4.23f represented an example of successful abnormal region detection by the four methods. Additionally, Fig. 4.23e the Esophagitis region was detected by all methods but the Faster R-CNN was able to provide a more localized detection compared to the ground truth annotation. For the rest of the Figs 4.23j to 4.23l which had a more challenging appearance, the Faster R-CNN, and SSD successfully picked the abnormal region while the Fast R-CNN and R-CNN failed to locate them. Moreover, the generated a false positive detection in Fig. 4.23j by detecting a region that had a different appearance than the rest of the area, but was not an abnormal region according to the ground truth.

The esophagus has a special internal structure that makes it challenging to differentiate between normal and abnormal regions. Also, the abnormalities in the esophagus are particularly challenging due to their different sizes, location, and shape. There exist variations in the size and the location in the generated bounding boxes from the four models, where each box might include non-cancerous regions. Table 4.8 calculated the average error presented by each model for both datasets in capturing non-cancerous regions inside the bounding box. As shown, the R-CNN and Fast

R-CNN presented a higher error percentage compared to the other two models. This indicates the bounding box generated by these two methods included a high ratio of non-cancerous regions. On the other hand, the Faster R-CNN and SSD provided a lower error rate for including non-cancerous areas, therefore they were able to provide better bounding boxes localized around the cancerous regions.

Table 4.8: Average error presented by each model in capturing non-cancerous regions inside the produced bounding boxes for both the MICCAI'15 and Kvasir dataset.

	R-CNN	**Fast R-CNN**	**Faster R-CNN**	**SSD**
Miccai'15	0.388	0.328	**0.201**	0.197
Kvasir	0.352	0.334	**0.181**	0.229

Throughout the evaluation; the Faster R-CNN and the SSD showed to have the leading performance regarding the different evaluation measures. However, we conclude that the Faster R-CNN is more convenient to use for esophageal abnormality detection as it performed better on the Kvasir dataset which is considered more challenging and has a larger number of data. Additionally, Faster R-CNN had fewer average error presented which indicates that the generated bounding box was more localized around the abnormal region.

Faster R-CNN with Gabor Features Results

In this section, experiments are carried out to evaluate the performance of the proposed method using each dataset separately. First, experiments are conducted to investigate the effect of extracting features based on the implemented DenseNet network. Then, we illustrate the effect of concatenating the Gabor features with CNN features on the detection performance. Moreover, we demonstrate different visual examples of the detection output from the utilized dataset using the proposed model. Finally, we compare the performance of the method with state-of-the-art results.

- **Evaluation of Esophagitis detection**

  In this section, we report the performance of our abnormality detection method in locating Esophgities regions. The Kvasir dataset was divided into 50% train-

ing, 10% validation and 40% testing by randomly selecting the images. First, to identify the effect of extracting features using DenseNet, we compare the detection results with the VGG'16 and AlexNet when used as a CNN backbone network for the Faster R-CNN. As mentioned earlier, the VGG'16 was used as the CNN backbone in the original Faster R-CNN. Table 4.9 displays the detection recall, precision, F-Measure, and mAP values when extracting CNN features with different CNN networks. As shown, extracting features using DenseNet improved the result of a recall by 4.3% & 5.2% and precision by 2.3% & 2.6% when compared to the other two networks. This implies that utilizing the Densenet to extract features enhances the information flow throughout the network with dense connections leading to improved performance.

Table 4.9: A comparison between different architectures as a backbone for the Faster R-CNN *DenseNet, VGG'16* and *AlexNet* evaluated on the Kvasir dataset.

Methods	**Recall** (%)	**Precision** (%)	**F-Measure** (%)	**mAP** (%)
DenseNet	**87.9**	**88.4**	**88.2**	**71.6**
VGG'16	83.6	86.1	84.8	68.9
AlexNet	82.7	85.8	84.2	67.2

Secondly, we compare the detection results after merging the Gabor features with the CNN features for the three networks. It can be seen from Table 4.10 that using the DenseNet with Gabor features was able to maintain the highest detection performance. Additionally, when comparing the results of Table 4.10 with Table 4.9, it can be concluded that adding the Gabor filter responses to the feature map enhances the texture information leading to an outstanding effect on the final results. As shown, the results of the detection were improved from 87.9% to 90.2% in the case of the DenseNet. Moreover, it had a positive impact on the other networks where the results were increased from 83.6% to 86.4% for VGG'16 and 82.7% to 86.1% for AlexNet. Furthermore, there is a 4.3% improvement in mAP by the proposed model compared to using the DenseNet only, which indicates a strong overall performance.

Table 4.10: A comparison of results after concatenation of the Gabor features with different CNN architectures as a backbone for the Faster R-CNN evaluated on the Kvasir dataset.

Methods	**Recall** (%)	**Precision** (%)	**F-Measure** (%)	**mAP** (%)
Proposed Model	**90.2**	**92.1**	**92.1**	**78.1**
VGG'16 Gabor features	86.4	89.1	87.7	74.2
AlexNet Gabor features	86.1	90.3	881	73.6

Moreover, we also plot the AP measure as a function of the IoU threshold in Fig. 4.25a. It can be observed that, for Esophagitis detection, the CNN network with the Gabor features outperform the network without the Gabor features. Also, our proposed model obtains a higher AP in a wider range of IoU threshold values than the other methods confirming the efficiency of our designed Densenet backbone network with Gabor features in the detection process.

Furthermore, Fig. 4.24 provides qualitative examples of our esophagitis detection results. Figs. 4.24a through 4.24f display samples of the images with correct detection. We find that our model can successfully detect various esophagitis regions of different sizes and appearances. The connection between preceding layers in DenseNet provides richer patterns. Therefore, the proposed model was able to detect small regions that were not detected by the other networks such as Fig. 4.24a, Fig. 4.24e & Fig. 4.24f. Moreover, in this study, if the generated bounding box has an intersection less than a threshold of 0.5 with the ground-truth (as described earlier) we consider the bounding box a false prediction, even though it correctly detected an abnormality (i.e. if the threshold had been set lower, the region would have been considered as TP), Fig. 4.24g & Fig. 4.24h illustrate examples of such cases. Moreover, Fig. 4.24i & Fig. 4.24j represent samples of the incorrect prediction. Most of the false predictions made by the model capture regions that have a differ-

ence in color/texture from the surrounding area. Additionally, Fig. 4.24k & Fig. 4.24l present negative outputs, as the detection model was not able to detect an abnormality in the endoscopic image. Overall, our model proved to have a strong performance in detecting esophagitis regions.

- **Evaluation of EAC detection**

The performance of the proposed model in detecting the EAC regions is reported in this section. For the MICCAI'15 dataset, we train and validate the model on LOPO-CV approach as the number of images from each patient is provided (i.e. LOPO-CV has the advantage of estimating less biased results). For the (LOPO-CV), that data is divided into N folds (*N is the number of patients*) where each fold excludes the full images of a single patient that is later used for testing and 10% of the fold is set aside for validation. First, we compare the proposed model with other CNN backbone networks for the Faster R-CNN as described in the previous section. Table 4.11 represents the results of the different CNN networks without Gabor features while Table 4.12 illustrates the results with Gabor features. From both Tables, the consequences of learning features with the DenseNet are presented by increasing the accuracy of detection by 5% & 7% with Gabor features and by 2% & 4% without Gabor features when compared with VGG'16 & AlexNet respectively. Additionally, the Gabor feature complements the feature map leading to a high recall rate in the detection of the EAC region correctly with fewer false regions. The superior performance of the proposed model is confirmed by comparing it with the other networks. As illustrated, adding the Gabor features increased the recall from 0.90 to 0.95, the precision from 88% to 91% and F-measure from 89% to 93% when using DenseNet as the backbone network. Also, in the case of using VGG'16 as a backbone network, the recall has increased from 0.88 to 0.90, precision from 86% to 87%, and F-measure from 0.87 to 0.88. In the case of using the AlexNet as backbone network recall has increased from 86% to 88%, precision from 87% to 88%, and F-measure from 86% to 88%.

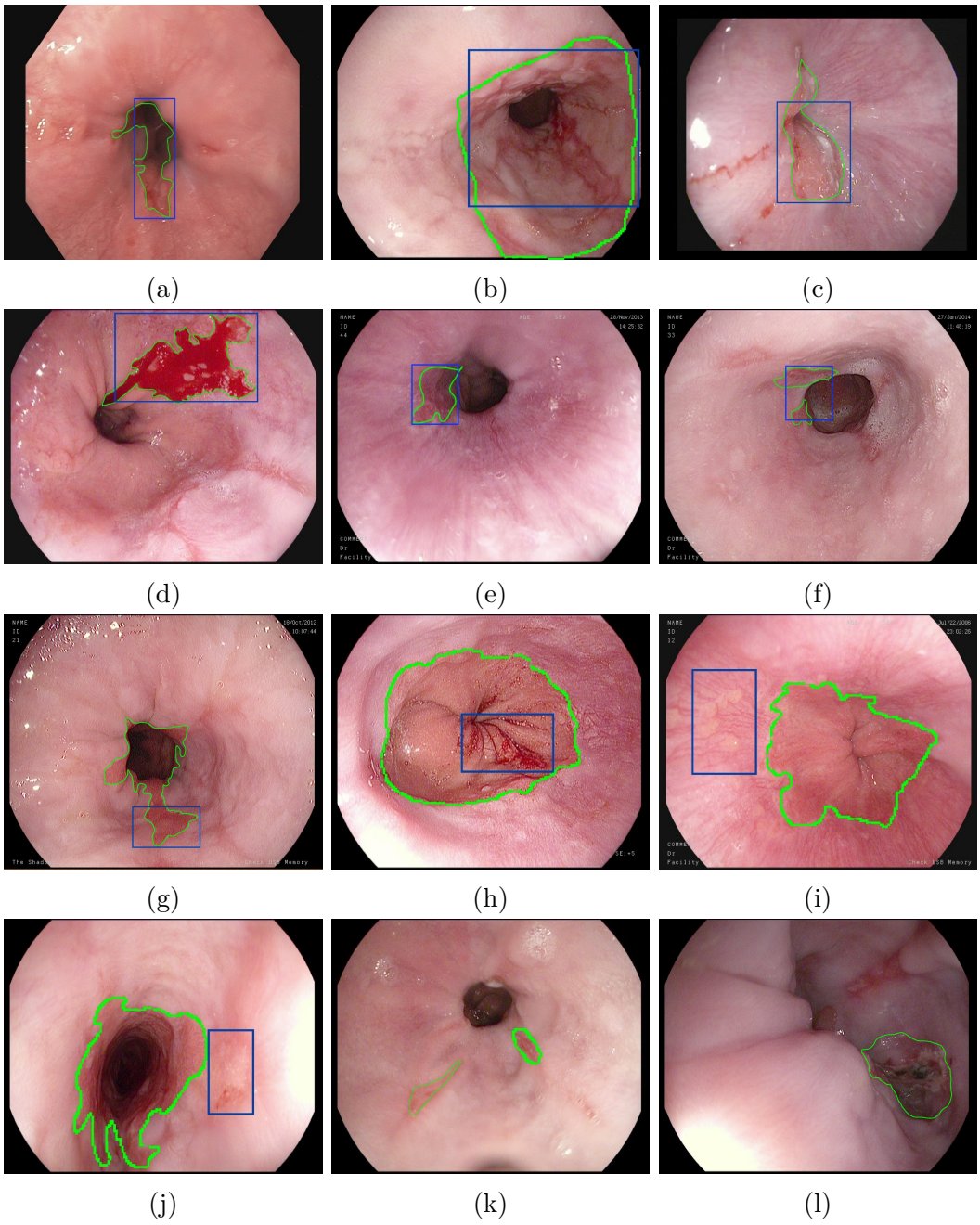


Figure 4.24: Detection examples from Kvasir dataset. The gold-standard by the expert is outlined with green lines in all the images. The generated bounding box by the model appears in the images with blue. From the first & second row, figures (a) to (f) represent correct detection results. Figures (g) to (j) represent samples some false predictions where (g) & (h) have an IoU<0.5 while (i) & (j) wrong locations. Figures (k) & (l) shows a false negative output where the model was not able to predict any abnormality.

Table 4.11: A comparison between different architectures as a backbone for the Faster R-CNN *DenseNet, VGG'16* and *AlexNet* evaluated on the MICCAI'15 dataset.

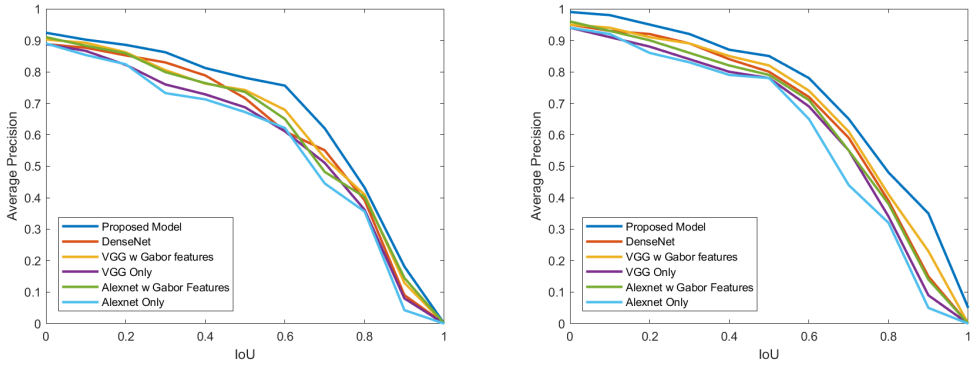
Methods	**Recall** (%)	**Precision** (%)	**F-Measure** (%)	**mAP** (%)
DenseNet	**90.0**	**88.0**	**89.0**	**81.0**
VGG'16	88.0	86.0	87.0	78.0
AlexNet	86.0	87.0	86.0	78.0

Table 4.12: A comparison of results after concatenation the Gabor features with different CNN architectures as a backbone for the Faster R-CNN evaluated on the MICCAI'15 dataset based on a LOPO-CV

Methods	**Recall** (%)	**Precision** (%)	**F-Measure** (%)	**mAP** (%)
Proposed Model	**95.0**	**91.0**	**93.0**	**85.0**
VGG'16 Gabor Feature	90.0	87.0	88.0	82.0
AlexNet Gabor Feature	88.0	88.0	88.0	79.0

Moreover, the mAP values have been increased from 81% to 84%. Fig. 4.25b represents the AP measure as a function of the IoU threshold for the MICCAI'15 dataset. As shown, the proposed model achieved a high AP over different IoU thresholds compared to the other networks, proving the effectiveness of the model in finding EAC regions.

To visualize the output from the proposed automatic detection method, we show examples for the correctly detected lesions, false positives, and missed EAC lesions in Fig. 4.26. As observed, the proposed method was able to successfully locate tumor regions in several EAC images. Examples for correct detection with challenging cases are shown in Figs. 4.26a to 4.26d. After inspecting the missed EAC lesions, we have found that most of the missed images are the tumors that mainly have a flat surface with the esophagus (for example; Fig. 4.26f). The false positives in our model are mainly images with high bar-



(a) Kvasir (b) MICCAI'15

Figure 4.25: AP-IoU threshold curves using different CNN network with and with Gabor features for Esophgities detection from Kvasir dataset and EAC detection in MICCAI'15 dataset.

rett's grade or have extreme changes in tissue color as shown in Fig. 4.26g & 4.26h.

- **Comparison with state-of-the-art methods**

  To validate the effectiveness of the proposed method, we compared the results of our detection method with the results of two state-of-the-art methods reported in (Van Der Sommen, F. Zinger S. et al., 2014) and (Mendel et al., 2017) that use the same dataset of MICCAI'15 to find EAC regions. For a fair comparison, the same validation method (LOPO-CV) is adapted. As shown in Table 4.13, the results of our detection methods outperformed the state-of-the-art methods in all evaluation measures with a *Recall: 95%, Precision: 91%,* and *F-measure: 93%*. Features learned using the proposed model achieved better results with reduced trainable parameters than (Van Der Sommen, F. Zinger S. et al., 2014) and (Mendel et al., 2017), demonstrating the effectiveness of reusing the features throughout the network and enhancing the model performance on the limited training data.

- **Additional Measures**

  The differences in recall and precision calculated using the proposed model and using the DenseNet without the Gabor features were statistically evaluated for both datasets, using the paired t-test at a confidence level of 95%.

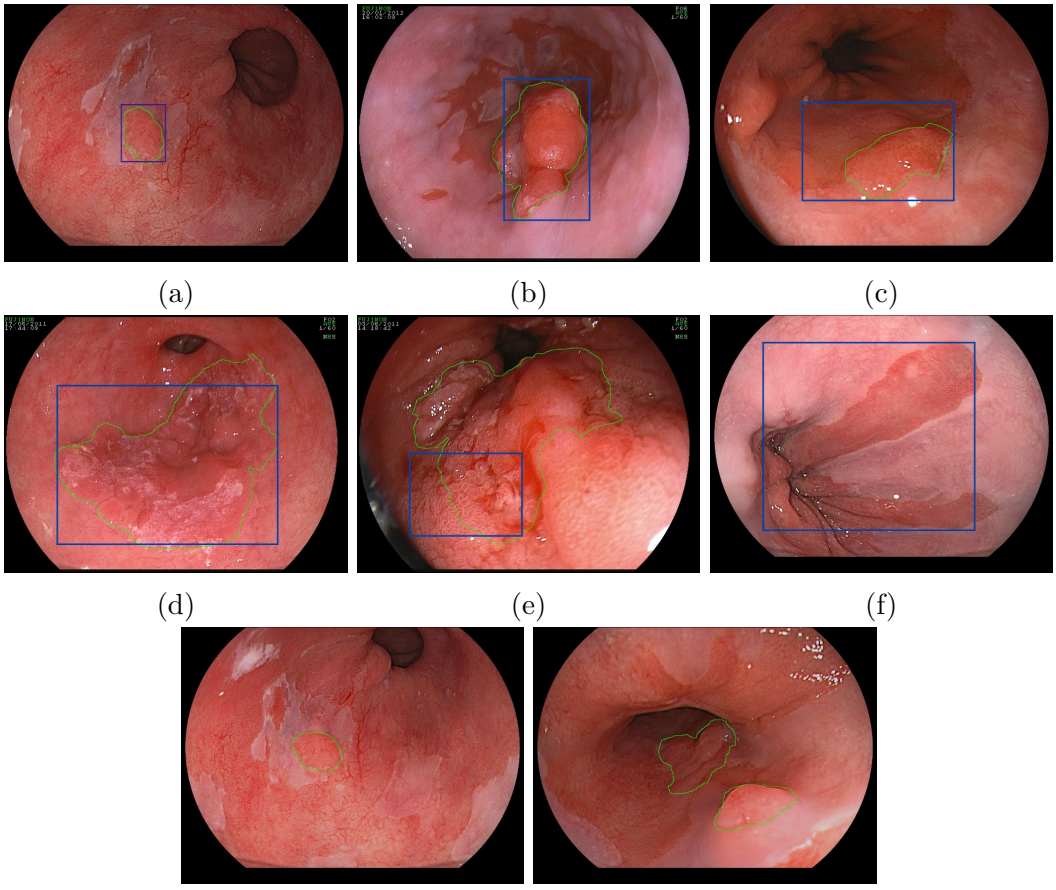


Figure 4.26: Detection examples from MICCAI'15 dataset, The gold-standard of the intersection between the 5 experts (sweet-spot region) is outlined with green lines in all the images. The generated bounding box by the model appears in the images with blue. The first row, from (a) to (d) represent correct EAC detection results. The second-row, (e) represents a false prediction (Intersection with ground truth $<$ 0.5 or wrong location), (f) false prediction in a non-cancerous patient and (g) & (h) both show a false negative output where the model was not able to predict any abnormality.

The results of the two-tailed *p-value* are provided in Table 4.14. For the Kvasir dataset, the difference between the recall and precision values for the proposed model was found to be significantly different when compared with the detection using features extracted by the DenseNet only. On the other hand, the MICCAI'15 dataset deemed to be significantly different only for the recall results. Moreover, the detection time during testing was also investigated. The average time to generate detection bounding boxes using our proposed model

Table 4.13: A comparison between the Proposed Model and state-of-the-art methods Sommen et al.(Van Der Sommen, F. Zinger S. et al., 2014) and Mendel et al.(Mendel et al., 2017) on the MICCAI'15 dataset based on a LOPO-CV

Methods	**Recall** (%)	**Precision** (%)	**F-Measure** (%)
Proposed Model	**95.0**	**91.0**	**93.0**
Sommen et al.	86.0	87.0	87.0
Mendel et al.	94.0	88.0	91.0

was an average of **2.34** seconds. We assume that the detection speed could be improved when using a more powerful GPU.

Table 4.14: The p-value calculated using the paired t-test to measure the difference of recall and specificity precision of proposed model with and without Gabor features on the two datasets

	Recall	**Precision**
Kvasir dataset	0.0055	0.00023
MICCAI'15 dataset	0.0447	0.10219

GFD Faster R-CNN Results

To represent the efficiency of the proposed GFD Faster R-CNN in improving the detection results we first compare it with results from the first model "Faster R-CNN with Gabor Features". Moreover, to illustrate the effectiveness of the GF feature fusion, we compare our model with the Faster R-CNN with only using the original endoscopic image features extracted by DenseNet. Additionally, to evaluate the impact of using the DenseNet as the backbone network, the results are also compared with the Faster R- CNN model using the VGG'16 (state-of-the-art Faster R-CNN) with and without fusing the GF features. Both Kvasir and Miccai'15 are used to evaluate our model.

- **Evaluation of Esophagitis detection**

The Kvasir data was divided randomly into 50% training, 10% validation and 40% testing. Table 4.15 yields a quantitative comparison of the detection performance in finding ***Esophagitis*** abnormalities with other Faster R-CNN networks. As shown, our proposed GFD Faster R-CNN outperformed against the other detection networks with a recall of 92.7%, precision of 94.2%, F-measure of 93.4% and mAP value of 82.4%. Precisely, the two-input network enhanced the overall detection performance when compared to the Faster R-CNN with the Gabor Features model, where the recall was increased by 2.5% and the precision by 2.1%. Moreover, the impact of GF features fusion with features from the original image is assessed. As shown in Table 4.15, when fusing the features, the performance of correctly detecting Esophagitis regions has improved the recall from 87.9% to 92.7% (using DenseNet) and from 83.6% to 89.2% (using the VGG'16). Moreover, the precision was enhanced from 88.4% to 94.2% (using DenseNet) and from 86.1% to 90.1% (using the VGG'16). The high recall and precision performances demonstrate that the fusion of the features provided rich feature representation that led to an improvement in the final detection stage. Furthermore, the detection results when using the DenseNet as the backbone network for feature extraction surpass the results when using the VGG'16. As illustrated, learning features using the DenseNet architecture increased the recall from 89.2% to 92.7% and the precision from 90.1% to 94.2% when fusing the GF features. Additionally, the results of the recall increased from 83.6% to 87.9% and the precision from 86.1% to 88.4% without considering the GF features. These results indicate the effectiveness of using the DenseNet as a backbone in providing a maximum flow of information that enhances the final detection results.

Samples of detection results are presented in Fig 4.27. The figures show samples of correct detection, false detection and missed regions (no prediction). If the generated bounding box overlaps with the ground-truth with less than 50% it is considered as a false detection even though it is in the correct area. As shown in Figs. 4.27a to 4.27f, the proposed model successfully detected the full abnormal region with different appearances and locations from the images. Samples

Table 4.15: Comparison of the GFD Faster R-CNN with other detection networks with/without GF features, using different backbones in detecting Esophagitis.

Methods	Recall (%)	Precision (%)	F-Measure (%)	mAP (%)
GFD Faster R-CNN	**92.7**	**94.2**	**93.4**	**82.4**
Gabor Features with Faster R-CNN	90.2	92.1	92.1	78.1
DenseNet Faster R-CNN	87.9	88.4	88.2	71.6
VGG'16 GF Faster R-CNN	89.2	90.1	89.6	75.2
VGG'16 Faster R-CNN	83.6	86.1	84.8	68.9

from false positives detection are also illustrated. The GF image was able to emphasize hidden details in the image which improved the overall feature representation leading to improved detection performance. In Fig. 4.27g, the detection is considered FP as the detection to not locate the full abnormal region and only a small region from it. Also, the model in Figs 4.27i & 4.27h captured region that had different properties than the normal region but it was still a wrong detection. Finally, Figs 4.27k & 4.27l represents samples from negative detection where the model failed to pick the abnormal regions. These experiments demonstrate the outstanding performance of our GFD Faster R-CNN model in detecting different esophageal abnormalities (Esophagitis).

- **Evaluation of EAC detection**

For the MICCAI dataset, the model was evaluated using this dataset based on LOPO-CV to detect **EAC** regions. Table 4.16 compares our performance with the state-of-the-art method Mendel *et al.* (Mendel et al., 2017) and the standard Faster R-CNN networks. The proposed GFD Faster R-CNN obtained a recall of 97%, a precision of 92%, an F-measure of 94% and an mAP value of 89%. Our method surpassed the state-of-the-art results in (Mendel et al., 2017) on the same dataset with the same validation method in terms of all performance measures. This illustrates that our method is more efficient than the patch-based CNN approach as suggested by (Mendel et al., 2017). Fur-

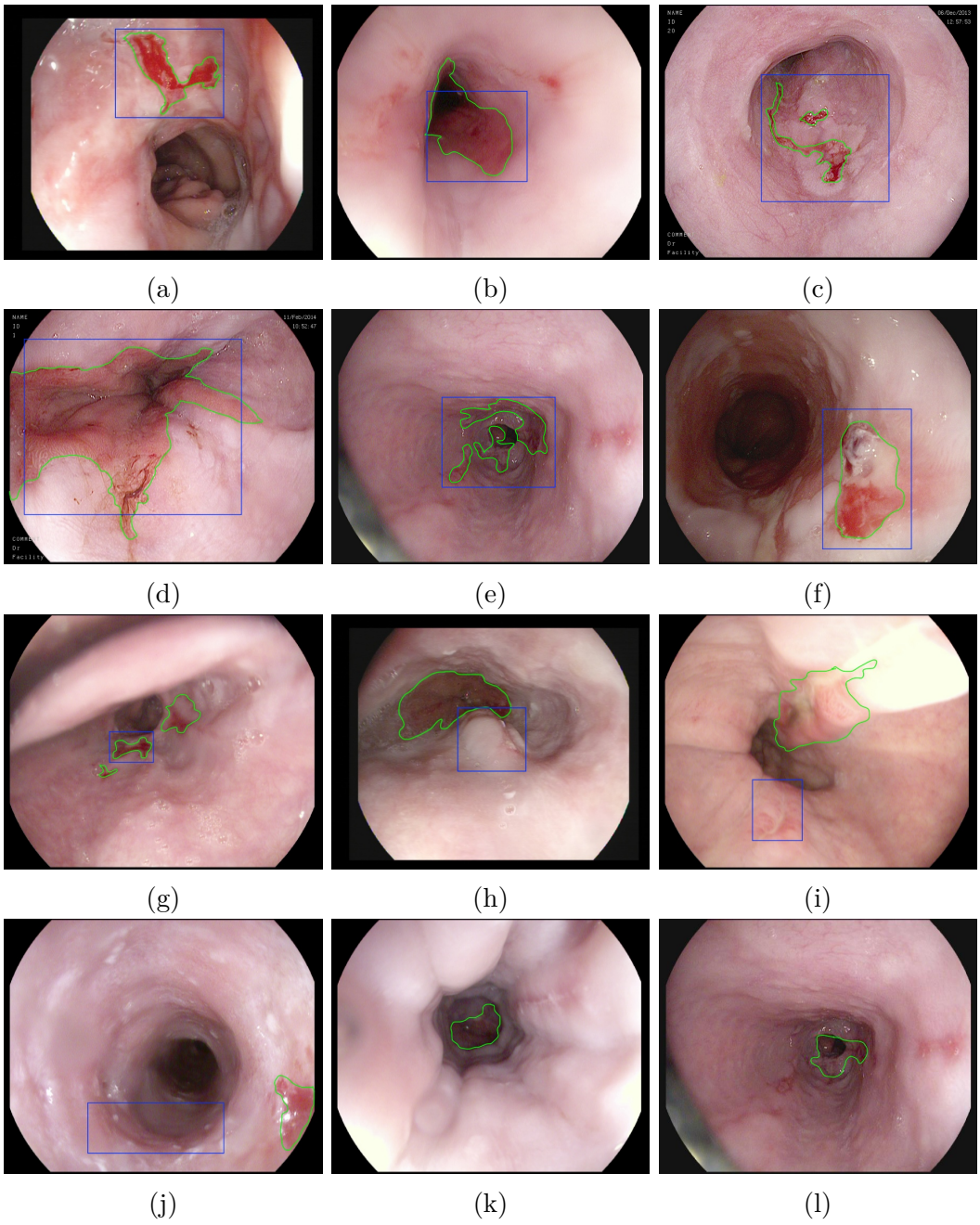


Figure 4.27: Examples of Esophagitis detection from the **Kvasir dataset** using GFD Faster R-CNN.. The gold-standard by the expert is outlined with green lines in all the images. The generated bounding box by the model appears in the images with blue. From the first & second row, figures (a) to (f) represent correct detection results. Figures (g) to (j) represent samples some false predictions where (g) & (h) have an IoU<0.5 while (i) & (j) wrong locations. Figures (k) & (l) shows a false negative output where the model was not able to predict any abnormality.

Table 4.16: Comparison of the GFD Faster R-CNN with other networks with/without GF features, different backbone networks and method by Mendel et al. (Mendel et al., 2017) to detect EAC.

Methods	**Recall** (%)	**Precision** (%)	**F-Measure** (%)	**mAP** (%)
GFD Faster R-CNN	**97.0**	**92.0**	**94.0**	**89.0**
Gabor Features with Faster R-CNN	95.0	91.0	93.0	85.0
DenseNet Faster R-CNN	90.0	88.0	89.0	81.0
VGG'16 GF Faster R-CNN	93.0	88.0	90.0	85.0
VGG'16 Faster R-CNN	88.0	86.0	87.0	78.0
Mendel et al.	94.0	88.0	91.0	——

thermore, by fusing the GF features, the results of detection are significantly improved when using different CNN backbone networks, increasing the recall from 90% to 97% and precision from 88% to 92% with the DenseNet and the recall from 88% to 93% and precision from 88% to 92% with VGG'16. It can be observed that using the DenseNet as a CNN feature extractor enhances the performance of the final detection. Additionally, the proposed GFD Faster R-CNN was able to further improve the performance when compared to the Gabor Features with Faster R-CNN results.

Moreover, Fig. 4.28 represents qualitative examples of the detection results from the MICCAI'15 dataset. Results show the exceptional performance of the model in locating EAC regions. Fig 4.28a to 4.28e shows positive detection examples of the cancerous region from different patients where the generated bounding box had a high IoU with the ground truth. In Fig. 4.28f the detection result is counted as a false positive because the model was only able to detect one abnormality region and missed the other one. A false detection is demonstrated in Fig. 4.28g, where, the bounding box was generated around a region that had a very distinctive difference in tissue color which may be the factor

of the error caused in detection. Additionally, 4.28h presented an example of negative output were the model was not able to located EAC regions.

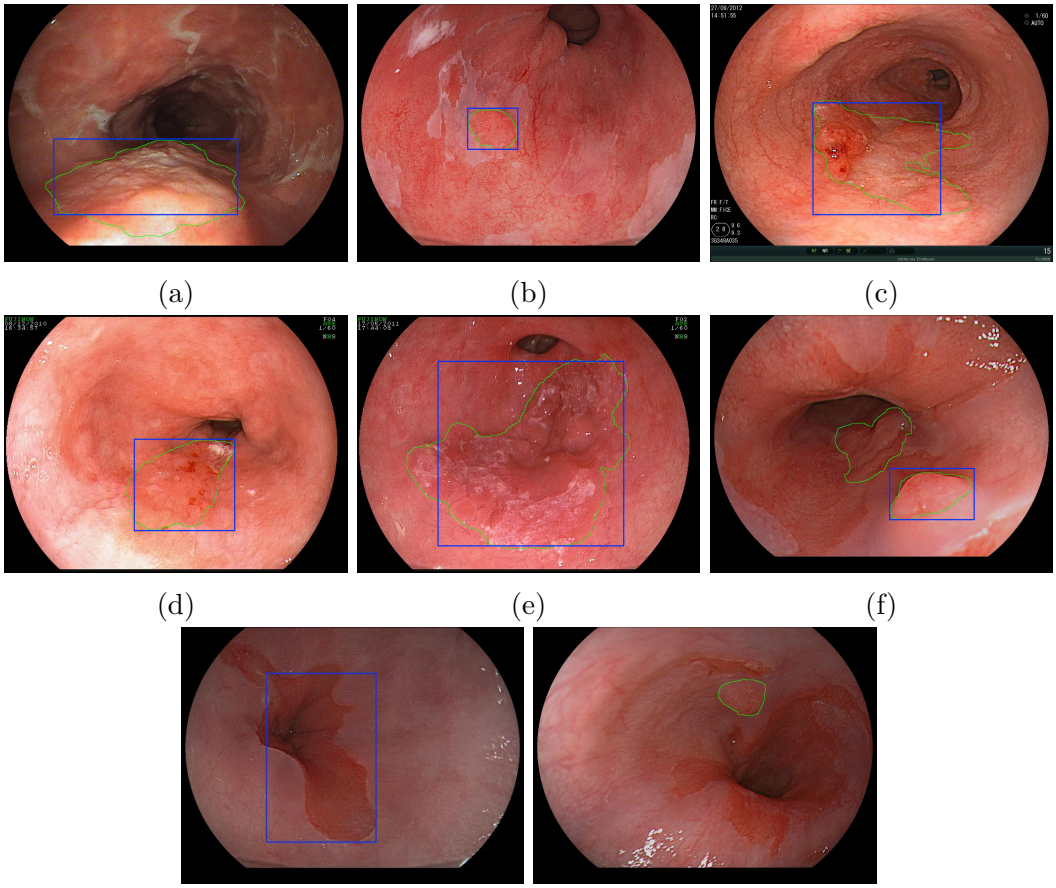
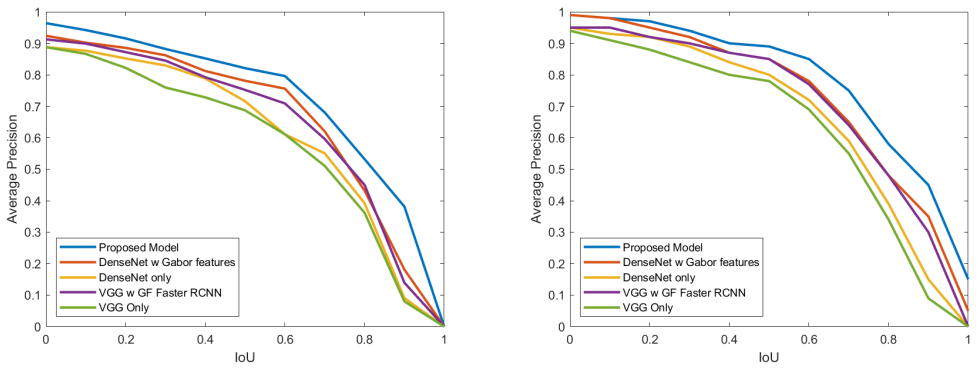


Figure 4.28: Examples of EAC detection from the **Miccai'15 dataset** using GFD Faster R-CNN, The gold-standard of the intersection between the 5 experts (sweet-spot region) is outlined with green lines in all the images. The generated bounding box by the model appears in the images with blue. Figures from (a) to (e) represent correct EAC detection results. Figures (g) represent a false prediction (Intersection with ground truth < 0.5 or wrong location), (f) false prediction in a non-cancerous patient and (h) show a false negative output where the model was not able to predict any abnormality.

Also, in Fig. 4.29 we plot the AP as a function of the IoU threshold for the results from both datasets. It can be concluded that the proposed model surpasses the other networks mentioned in Table 4.15 and 4.16 proving the effectiveness of the designed network in the process of detection. Additionally, we include the results from the Faster R-CNN with the Gabor features method in the curve comparison. As shown,



(a) Kvasir (b) MICCAI'15

Figure 4.29: AP-IoU threshold curves using the GFD Faster R-CNN (i.e. Proposed Model) and compared with other networks

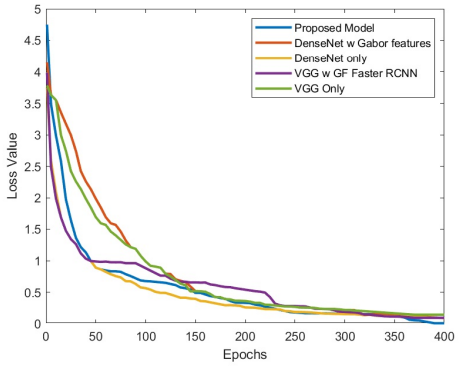
the GFD Faster R-CNN obtains a higher AP confirming its outstanding performance on different ranges of the IoU.

Furthermore, In Fig. 4.30 we plot the loss curves versus the number of epochs when training the different for both datasets (i.e. Kvasir (Fig. 4.30a) and MICCAI'15 ((Fig. 4.30b)). In addition to the training accuracy versus the number of epochs for both datasets i.e. Kvasir (Fig. 4.30c) and MICCAi'15 ((Fig. 4.30d)).

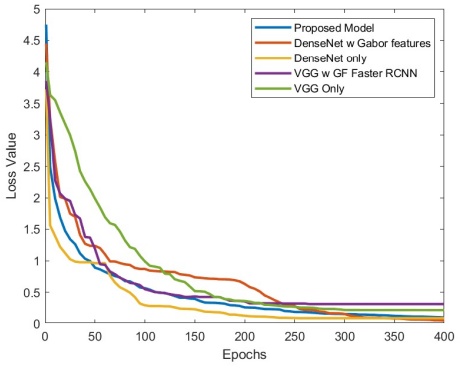
4.7 Summary

In this chapter, a deep learning method to automatically detect esophageal abnormalities from endoscopic images is presented. The proposed methods were evaluated on two publicly available datasets MICCAI 2015 (*Sub-Challenge Early Barrett's cancer detection* n.d.) and KVASIR (Pogorelov et al., 2017).

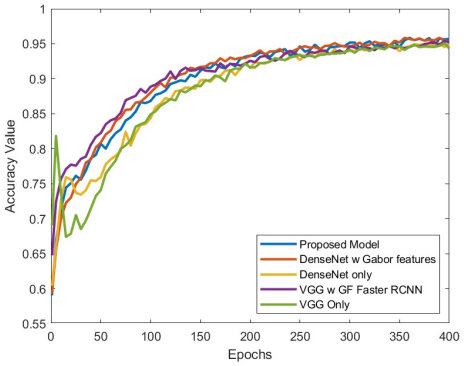
First, we adapted the state-of-the-art deep learning object detection methods to automatically identify the abnormalities from esophageal images. Throughout the evaluation experiments; the Faster R-CNN and SSD have proved to be the leading performers regarding the different evaluation measures, with an outstanding results of recall (*0.88, 0.96*), precision (*0.88, 0.96*), and F-measure (*87%, 94%*) respectively for the MICCAI'15 dataset based on LOPO-CV . While for the Kvasir the dataset, the result of recall (*83.6%, 80.1%*), precision (*86.1%, 78.4%*), and F-measure (*84.8%,*



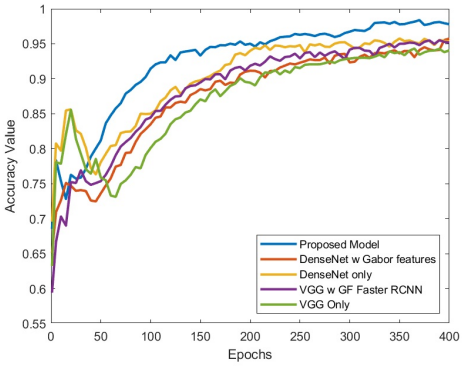
(a) Kvasir Loss Curve Vs. Epochs



(b) MICCAI'15 Loss Curve Vs. Epochs



(c) Kvasir Acc. Curve Vs. Epoch



(d) MICCAI'15 Acc. Curve Vs. Epoch

Figure 4.30: Loss curves Vs. Epoch and Accuracy curves Vs. Epoch when training both datasets Kvasir and MICCAI'15 using the GFD Faster R-CNN (i.e. Proposed Model) and compared with other networks.

79.2%) for the Faster R-CNN and SSD respectively. Although the results from the Faster R-CNN and SSD are considered comparable, however, the Faster R-CNN was able to generate Bounding Boxes that locate the abnormal regions with higher IoU with the Ground Truth (GT).

Secondly, a hybrid learning-based method was proposed which integrates handcrafted features with CNN features to automatically detect abnormalities. The Gabor filter responses calculated from endoscopic images are incorporated into the Faster R-CNN while adopting the DenseNet as the backbone network for CNN feature extraction. The dense connectivity in DenseNet improves the flow of information and the efficiency of parameters throughout the network by reusing learned features from the previous layers. The Gabor features proved in the literature its ability in detecting intestinal juices and providing related features regarding the esophageal

cancerous regions. Experimental results demonstrate that the fusion between the extracted Gabor features and the CNN features has improved the information used by Faster R-CNN for abnormality detection. Our newly designed architecture is validated on two datasets (Kvasir and MICCAI 2015). Regarding the Kvasir, the results show an outstanding performance with a recall of 90.2% and a precision of 92.1% with a mean of average precision (mAP) of 75.9%. While for the MICCAI 2015 dataset, the model was able to surpass the state-of-the-art performance with 95% recall and 91% precision with mAP value of 84%. Experimental results show that the system can detect abnormalities in endoscopic images with good performance without any human intervention.

Finally, to further improve the results, a novel GFD Faster R-CNN network that automatically detects esophageal abnormalities from endoscopic images is proposed. A significant effort has been made to adapt the Faster R-CNN to address the challenges of esophageal abnormality detection which includes the generation of GF image and employing the DenseNet to learn discriminative features from both endoscope and GF images. The GF image is produced by maximizing each pixel value based on different Gabor filter responses of the endoscopic image, resulting in an enhanced image that highlights the hidden fractal details. The RPN layer suggests region proposals for the candidate region using only CNN features extracted from the original endoscopic image. Features extracted from the GF and endoscopic images are fused through bilinear fusion before the ROI pooling stage in Faster R-CNN, providing a rich feature representation that boosts the performance of final detection. Extensive experiments have been carried out to evaluate the performance of the model, with a recall of 0.927 and a precision of 0.942 for Kvasir dataset, and a recall of 0.97 and a precision of 0.92 for MICCAI'15 dataset, demonstrating a high detection performance compared to the state-of-the-art.

Furthermore, the difference in the results between the Faster R-CNN with Gabor features and the GFD Faster R-CNN methods were statistically evaluated using the two tailed t-test to validate the difference. The results were found to be significantly different with a confidence level of 5% (p-value$<$ 0.05).

All the models were trained and evaluated on two different datasets. An additional advantage of the proposed methods is that it is trained using the full image as an input instead of patches from the image as used by other methods in the literature (Mendel et al., 2017). The results of this work have been published as follows: (1) The evaluation of deep learning methods has been published in International Journal of Computer Assisted Radiology and Surgery (IJCARS) (N. Ghatwary, Zolgharni and Ye, 2019a), (2) The Faster R-CNN model with handcrafted Gabor features has been published in the Journal of IEEE Access (N. Ghatwary, Ye and Zolgharni, 2019) and (3) The GFD Faster R-CNN has been published in the workshop Machine Learning in Medical Imaging (MLMI) (N. Ghatwary, Zolgharni and Ye, 2019b) that is held in International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI).

Future studies are likely to investigate the abnormality detection from endoscopic videos (i.e. instead of selected images) with more types of abnormalities (such as BE and SCC). The process of detection from the videos is considered more challenging than images due to the video properties. The process of automatic detection from endoscopic videos will be further investigated in the following chapter (Chapter 5).

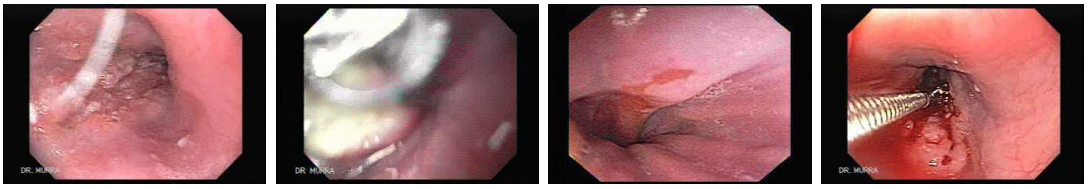
Chapter 5

Esophageal Abnormality Detection from Endoscopic Videos using Deep Learning

5.1 Introduction

In the previous chapter, an automated detection method was proposed that detects esophageal abnormalities from still frames. The results presented a promising performance compared to the state-of-the-art methods for detecting different types of abnormalities on different datasets. However, when the detection model developed based on the still frames (i.e. 2D networks) was applied to the full video (as will be shown in this section in Table 5.2), it did not present good performances. The reason behind this is that CNNs are highly sensitive to small changes, disturbances and noises as shown in different recent studies (Moosavi-Dezfooli, A. Fawzi, O. Fawzi et al., 2017; Su, Vargas and Sakurai, 2019; Narodytska and Kasiviswanathan, 2017; Nguyen, Yosinski and Clune, 2015; Moosavi-Dezfooli, A. Fawzi and Frossard, 2016).

Jiawei Su *et al.* (Su, Vargas and Sakurai, 2019) has proved that current DNNs are vulnerable to small changes and can easily be misled by adding just one pixel. Therefore, CNN networks might be distributed by the small changes in esophageal abnormalities appearance in endoscopy. This indicates that CNN networks can easily miss the same esophageal abnormal region appearing in a sequence of neighboring frames and produce unstable detection output with a high number of FPs. Addi-



(a) (b) (c) (d)

Figure 5.1: Examples of challenges frames from esophageal endoscopic videos. (a) Low-quality image, (b) blurred image, (c) challenging appearance, (d) Tool appearance.

tionally, the process of automatic detection from videos is an extremely challenging task. The rapid movement of the endoscope produces low-quality and blurry images (Fig 5.1a & Fig. 5.1b). Moreover, the endoscope is not always centered on the examined region with a limited abnormality view inside the esophagus (Fig. 5.1c). Furthermore, the occurrence of intestinal juices and tool appearance can block the presence of the abnormality (Fig. 5.1d). Therefore, detecting abnormalities from the esophageal endoscopic videos is very different from detecting the abnormalities from selected images.

All the previous work put much effort into studying different handcrafted features, conventional classifiers and DNNs to find suitable models for detection abnormalities from selected still frames/ images. No work in the literature has focused on finding the different types of esophageal abnormalities from videos or a sequence of frames. However, deep learning methods have been investigated for the automatic detection of polyps from colonoscopic videos (i.e. lower GI tract) (Du et al., 2019). Recent studies provided temporal information with spatial information from colonoscopy videos as additional feature representation for more accurate polyp detection (Chao, Manickavasagan and Krishna, 2019). A two-step approach was introduced by (Tajbakhsh, Gurudu and Liang, 2015), the method first extracts geometric features: *color, texture clues* and *shape context* to detect candidate regions. Afterward, several 2D-CNN are used to learn features surrounding each candidate region from the current, preceding and successive frames to learn the polyp spatial-temporal patterns. Later studies confirmed that the 3D network designs are more suitable for the video datasets (Misawa et al., 2018). Yu et al. (L. Yu et al., 2016) presented 3D fully convolutional network (3DFCN) to detect polyps in colonoscopic videos while redu-

cing FPs. The 3D-CNN extract spatiotemporal features by performing convolutions along the width, height, and temporal dimensions. This proved the capability of 3D-CNN to learn more illustrative spatiotemporal features from colonoscopic videos compared to 2D networks. Moreover, other methods made use of temporal dependencies between a consecutive set of video frames to provide useful information in detecting polyps when combined with spatial information (Zhang et al., 2018; Qadir et al., 2019).

In computer-aided diagnosis, the high precision and recall results are important to provide accurate detection analysis. In the discussed methods, the extracted spatiotemporal features showed its effectiveness when incorporated in the model by producing a high precision value, but with low recall value that needed to be improved. Therefore, the possibility of adjusting the spatiotemporal features with the appropriate model should be investigated to improve the overall detection performance.

Recently, Convolutional Long Short-Term Memory (ConvLSTM) (i.e. type of Recurrent Neural Networks (RNN) that will be described first in this chapter (Sec. 5.2)). The ConvLstm can learn the spatiotemporal consistency across the surgical video frames (Nwoye et al., 2019) and preserve the spatiotemporal regularity between neighboring frames (H. Zhu, Vial and Lu, 2017). Studies proved the efficiency of ConvLstm in learning the temporal variable characteristics from the sequence of frames when incorporated in deep learning networks (Mathai, Gorantla and Galeotti, 2019). When the ConvLstm is included with 3D-CNN, the network covers the short-term temporal information and long-term temporal information along with spatial information; producing a feature map that covers the spatiotemporal features of a longer sequence of video frames (G. Zhu et al., 2018; J. Huang et al., 2018).

In this chapter, we propose a novel 3D Sequential Dense-Convlstm Faster R-CNN for the detection of esophageal abnormalities (cancerous and precancerous) from endoscopic videos. The network is built using the concept of the densely connected convolutional network (DenseNet) (G. Huang et al., 2017), which propagates the gradient and feature information throughout the network by taking each layer as

input for all its upcoming layers. In our model, the DenseNet has been modified in several aspects where we propose increasing the internal layers of the dense blocks in the network sequentially to provide more related information. Moreover, for the construction of the network, we utilize the 3D-CNN with ConvLstm (Xingjian et al., 2015) to extract spatiotemporal features. The 3D-CNN extracts features regarding the third dimension (i.e. *Time*) holding richer information while the ConvLstm explores the relation of spatiotemporal information between video frames. The architecture of the proposed network is designed to extract features from videos and allows each frame to learn features from subsequent frames. Moreover, the extracted features are then used by the Faster R-CNN to generate bounding boxes to locate the abnormalities throughout the video.

To improve the overall detection performance, we propose a post-processing method named Frame Search Conditional Random Field (FS-CRF). The proposed FS-CRF employs a frame search algorithm with the Dense CRF on a frame-based level to improve the performance by recovering missed regions and removing false positives. To validate the efficiency of our proposed method, we evaluate it on a large dataset that is composed of different types of abnormalities. The contributions of this chapter can be listed as follows:

- An effective approach for the detection of different types of esophageal abnormalities (BE, EAC and SCC) from endoscopic videos is presented. To the best of our knowledge, this is the first approach to detect different types of esophageal abnormalities using the full videos instead of selected frames.
- We design a novel 3D Sequential Dense-Convlstm backbone network to extract features from esophageal endoscopic videos. By incorporating the 3D-CNN and the ConvLstm, the proposed network has the ability to learn spatiotemporal features that is more compatible with the properties of videos.
- We implement an FS-CRF for post-processing that can recover the missed abnormal regions in a sequence of consecutive frames based on the initial detection output to improve the overall detection performance.
- We extensively validate the proposed model using endoscopic videos dataset

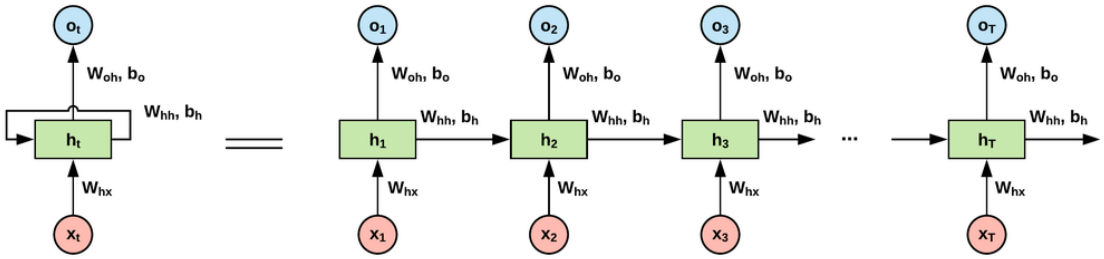


Figure 5.2: The standard process of an RNN layer (Olah, 2015)

that includes normal, precancerous and cancerous patients. Moreover, we compare the performance of the model with different types of networks and datasets such as the 2D-CNN network and colonoscopy dataset.

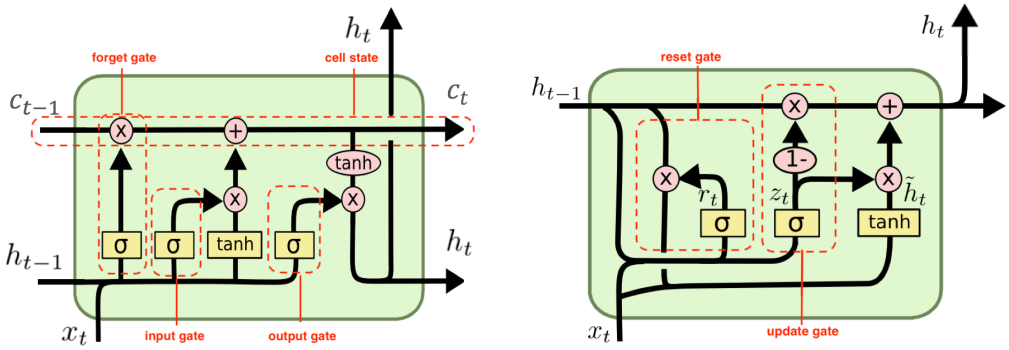
5.2 Overview of Recurrent Neural Networks (RNN)

RNN is a type of neural networks that is designed to deal with a series of inputs (i.e. temporal sequence) (Karpathy, Johnson and Fei-Fei, 2015). The main concept of the RNN, that it has the ability to learn from the past using a **memory** that provides information to the next layer. Therefore, the output from the RNN network is not only determined by the input but also from the complete history of input. As shown in Fig. 5.2, the RNN can be considered as a repeated copy of the same network that pass a message to the successor and mainly composed of two terms: the *hidden state* (h_t) and the *current input* (x_t). The tanh (eq. 4.3) is used as the activation function for the RNN to ensure that the values of the output stay in the range of $-1 < x < 1$.

The problem of the RNN that it has a short-term memory, therefore, it is not suitable with long sequence problems. Morover, it faces the problem of the vanshing gradient. To solve these issues, RNN units were proposed: *LSTM* and *GRU*. These units have an internal mechanism known as **gates** that manage the flow of information.

- **Long Short-Term Memory (LSTM)**:

  The LSTM (Hochreiter and Schmidhuber, 1997) is made of different gates as represented in Fig. 5.3a that characterize the data into short-term and long-term. These gates help the RNN to decide which data is important to be passed to next layer and what can be discarded. The LSTM gates are:



(a) LSTM (b) GRU

Figure 5.3: A representation of the internal model operation for the LSTM and GRU (Olah, 2015).

- *Input Gate (i_t) :* The input gate is responsible to update the cell state (described next). First the hidden state (h_t) and current input (x_t) are passed to a (σ) function so their values become in range from 0 to 1 to decide its importance. Additionally, to help regulate the network the values of h_t and x_t are passed to a tanh function.
- *Forget gate (f_t):* This gate is responsible to decide what information should be ignored and what should be used from previous input. This gate utilizes the sigmoid function (eq. 4.2) to get a value in the range $0 < f(x) < 1$. If the value is nearer to 0, therefore, it is forgotten and if it is closer to one then it is kept.
- *Cell State (c_t):* In the cell state, some values are dropped if the output of it is near to 0 when it is multiplied with the forget vector. The output from the input gate (i_t) is exposed to a point-wise addition that updates the cell state with new values.
- *Output Gate (o_t):* The output gate is responsible to determine the next hidden state.

Each of these gates are represented by the following equations:

$$i_t = \sigma_g(x_t U^i + c_{t-1} W^i + b_i) \tag{5.1}$$

$$f_t = \sigma_g(x_t U^f + c_{t-1} W^f + b_f) \tag{5.2}$$

$$o_t = \sigma_g(x_t U^o + c_{t-1} W^o + b_o) \tag{5.3}$$

$$c'_t = tanh(h_{t-1} U^c + x_t W^c + b_c) \tag{5.4}$$

$$c_t = \sigma(f_t \circ C_{t-1} + i_t \circ c'_t) \tag{5.5}$$

$$h_t = tanh(ct) \circ o_t \tag{5.6}$$

where, i_t, f_t, o_t and c_t represent the input gate, forget gate, output gate and the cell gate respectively at time-step t. The h_t is the hidden state vector that represents the output from the LSTM unit. Moreover, the $\circ$ denotes the element-wise product operation.

- **Gated Recurrent Unit (GRU)**:
  GRU is the most recent RNN unit (K. Cho et al., 2014). The concept of the GRU is very similar to the LSTM but with the only two gates as shown in Fig. 5.3b. The complexity of the GRU is considered more efficient as it has fewer gates and operations compared to the LSTM. The GRU gates are:
  - *Reset Gate:* The reset gate is used to select the amount of past information to be kept and decide how to combine the new input with it.
  - *Update Gate:* The update gate performs similarly to the forget and input gate of the LSTM. It selects the information to be thrown away and the new information to add.

  If all the reset gates are set to $(1's)$ and the update gates are set to $(0's)$ thereby it becomes a standard RNN model. The operation units of the GRU are described as follows:

$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \tag{5.7}$$

$$z_t = \sigma(W^z x_t + U^z h_{t-1}) \tag{5.8}$$

$$h'_t = tanh(W^h x_t + U^h h_{t-1}) \tag{5.9}$$

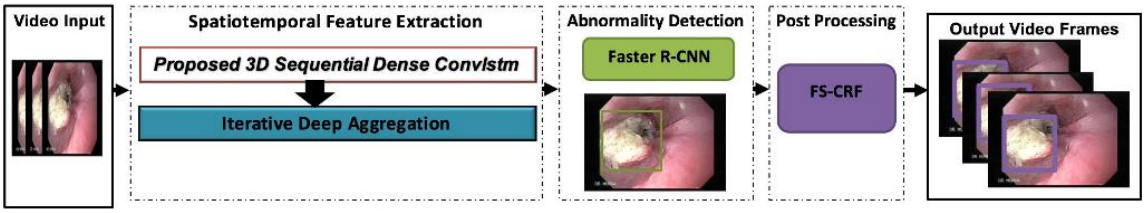


Figure 5.4: Overview of the abnormality detection approach. First, Spatiotemporal features are extracted from the video input using the proposed 3D Seq. DenseConvLstm network. Secondly, these features are used by Faster R-CNN to generated BBs for EAC regions in the video. Finally, a novel Post-Processing approach is applied for final video detection output.

$$h_t = z_t \circ h_{t-1} + (1 - z) \circ h'_t \tag{5.10}$$

where, r is rest gate and z is the updated gate at time step t. The h_t is the hidden state vector that represents the output from the GRU unit. Moreover, the $\circ$ denotes the element-wise product operation.

5.3 Methodology

In Fig. 5.4 we illustrate the proposed automatic detection framework which involves three main stages: *(i) spatiotemporal feature extraction, (ii) detection of abnormality regions* and *(iii) post-processing phase*. As shown, first the spatiotemporal features are extracted using a novel 3D Sequential DenseConvLstm Network that is equipped with dense connectivity and integrated with both 3D-CNN and ConvLstm. The integration between the feature extracted from 3D-CNN and ConvLstm preserves the global temporal connectivity between subsequent frames. Moreover, we set a ConvLstm layer to be the initial filter for the video input. Afterwards, the extracted features from each dense block are aggregated together. Then the extracted spatiotemporal features are used by the region proposal network (RPN) and region-of-interesting pooling layer (ROI Layer) in the Faster R-CNN to detect the region of abnormalities in the video frames. Finally, the detection results are post-processed with a proposed FS-CRF to improve the final performance of the model. In the remainder of this section, we explain each stage with its components in detail.

5.3.1 Spatiotemporal Feature Extraction: 3D Sequential Dense-ConvLstm

The network is built on the concept of DenseNet architecture (G. Huang et al., 2017), which encourages feature reuse by connecting the output of a layer to all upcoming layers in the network. The proposed 3D Sequential Dense-ConvLstm is made up of three main components: *Sequential Dense Block*, *SpatioTemporal Transition Layer* and *Growth Rate*. Features extracted from each *Sequential Dense Block* are combined through *Iterative Deep Aggregation* to build the feature map used in the next stage.

Sequential Dense Block (Seq-DB)

The **DB** performs the operation of the dense connectivity where each layer takes the feature map of all previous layers as an input. The output of a DB is given by:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]) \tag{5.11}$$

where x_l is the feature map output from each DB with l layers. The $H(.)$ represents the composite operation formed of Batch Normalization (BN), Relu, and Convolutional operation. The (.) denotes the concatenation process inside the DB block where they must have the same size (i.e. height and width).

For the **Seq-DB** we propose two contributions; first, we increase the number of internal layers sequentially, therefore, the l layers per block are equal to the DB position in the network as shown in Fig. 5.5. For Simplicity, the output for block Seq-DB$_N$ is x_N where $l = N$ (i.e. $x_N = H_l[x_0, x_1, ..., x_{N-1}]$). The DenseNet depends more on high-level features than low-level ones, therefore, increasing the features in later DB's will provide more global features. Accordingly, it can maintain high performance with a reduced number of trained parameters.

Secondly, we propose using 3D-CNN as an operation in the composite function to learn local spatiotemporal features instead of 2D-CNN as in the original densenet. The operation of each layer (l) in Seq-DB is: (BN, Relu, & (3×3×3) 3D-CNN).

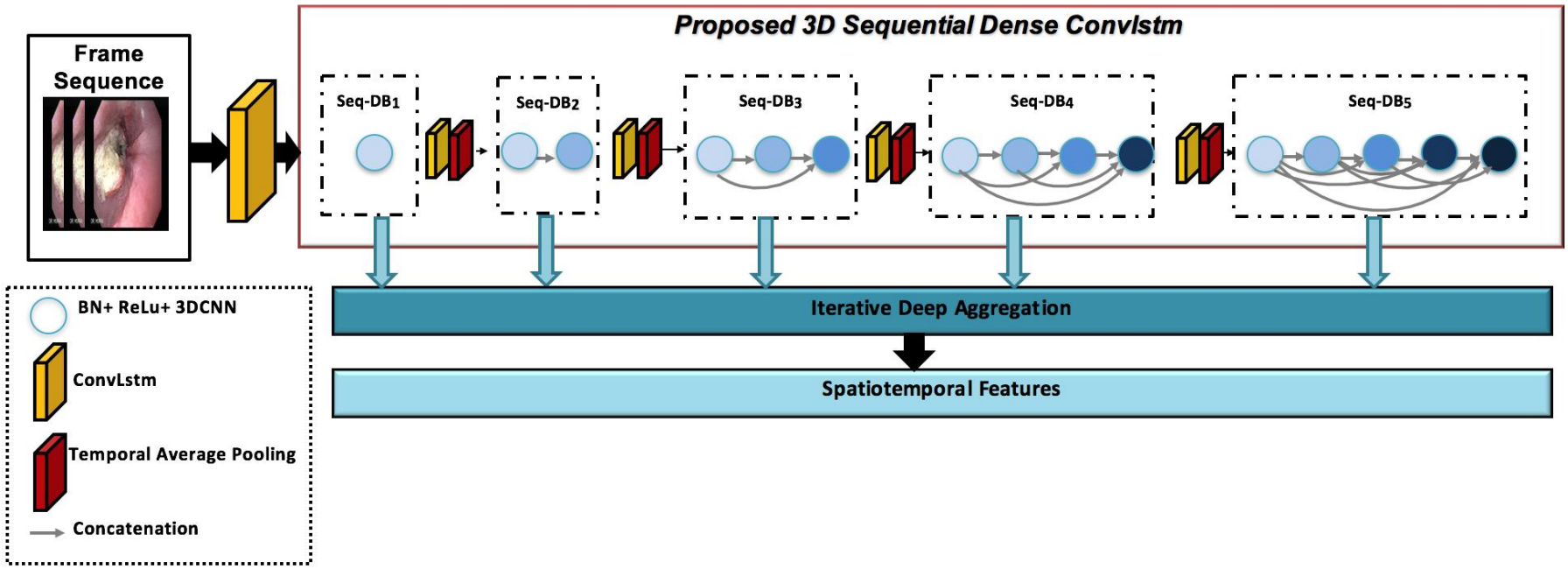


Figure 5.5: Overview of the proposed 3D Sequential Dense-ConvLstm network to extract spatiotemporal features from the video input. An initial ConvLstm layer is applied to input video then the Sequential Dense-ConvLstm is composed of Seq-DB blocks with Seq-TL in between. Finally, the output features from each Seq-DB are iteratively aggregated together.

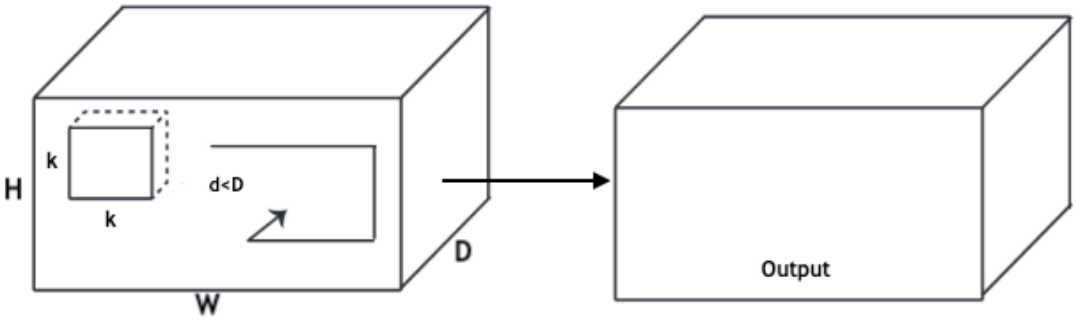


Figure 5.6: Applying 3D convolution in $W \times H \times D$ video volume with $k \times k \times k$ kernel results in another volume. The three dimensions represent width (W), height (H) and temporal dimension (D) respectively.

- **3D Convolution Neural Network (3D-CNN) Unit**: The 3D-CNN has the ability to extract short-term temporal features along with the spatial information, therefore, it is useful to use with the video analysis (Akilan et al., 2019). For the 3D-CNN, the operation convolves a 3D filter therefore both the feature map and kernel have a depth dimension (i.e. spatial dimension and a temporal depth). The 3D convolution kernel shares the respective channels at the time of execution and also between N consecutive frames. The process of 3D-CNN is shown in Fig. 5.6 and computed as follows:

$$3DC(x, y, z) = \sum_{d=0}^{D-1} \sum_{w=0}^{W-1} \sum_{h=0}^{H-1} k(d, w, h) * F(x + d, y + w, z + h) \tag{5.12}$$

where $*$ denotes the convolution operation, D is the depth, W and H are the width and height of the Kernal k. The $\{x, y, z\}$ and $\{d, w, h\}$ represent the coordinates of the input and the element index respectively.

SpatioTemporal Transition Layer (ST-TL)

The **ST-TL** exists between each DB which helps downsample the feature map. In the network, the proposed ST-TL is composed of (1×1×1) *ConvLstm* with stride=$(1 \times 1 \times 1)$ and same-padding to maintain the size of spatiotemporal feature map. Also, a (3×3×3) temporal average pooling with stride=$(2 \times 2 \times 2)$ is applied after the ConvLstm layer.

- **Convolution Lstm (ConvLstm) Unit:** ConvLstm is a type of Recurrent neural networks (RNNs) (Xingjian et al., 2015). RNNs have proved the ability to learn temporal information in several fields (Karpathy, Johnson and Fei-Fei, 2015). Precisely, Convolution Long Short Term Memory (ConvLstm) is a development of the LSTM with a convolution operation inside the LSTM cell (i.e. explained in Sec. 5.2). ConvLstm uses convolution operation instead of matrix multiplication at each gate of the LSTM cell. The ConvLSTM is designed for 3-D data as its input (i.e. such as videos) while LSTM input data are one-dimensional.

  The ConvLstm is capable of learning long-term spatiotemporal information by encoding the changes of spatial and temporal information using the convolution gates in it. Additionally, the convolution operations capture the underlying spatial features of multiple dimension data. Moreover, it forces consistency across time by taking into account the features from previous frames leading to improved detection. The operations of convolution and recurrence of the input-to-state and state-to-state transitions benefit from the spatiotemporal correlation information. The ConvLstm can be expressed as:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \odot c_{t-1} + \hat{b}_i) \tag{5.13}$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{ci} \odot c_{t-1} + \hat{b}_f) \tag{5.14}$$

$$o_t = \sigma((W_{xo} * x_t + W_{ho} * h_{t-1} + W_{ci} \odot c_t + \hat{b}_o) \tag{5.15}$$

$$g_t = tanh((W_{xc} * x_t + W_{hc} * h_{t-1} + \hat{b}_c) \tag{5.16}$$

$$ct = f_t \odot c_{t-1} + i_t \odot g_t \tag{5.17}$$

$$ht = o_t \odot tanh(c_t) \tag{5.18}$$

  where $*$, σ and $\odot$ denotes the convolution operation, sigmoid function, and entry-wise product respectively. At time step t, i_t, f_t, o_t, g_t, h_t are input gate, forget gate, output gate, modulation gate and hidden gate. The c_t is the sum of previous memory cell c_{t-1} which is modulated by f_t and g_t. Additionally,

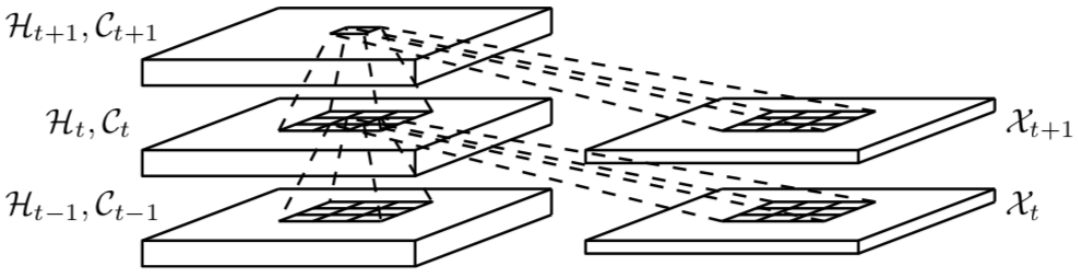


Figure 5.7: The inner structure of ConvLSTM (Xingjian et al., 2015)

W_{xi}, W_{xf}, W_{xo}, W_{xc} and U_{hi}, U_{hf}, U_{ho}, U_{hc} are the 2D convolutional kernels with biases $\hat{b}_i$, $\hat{b}_f$, $\hat{b}_o$ and $\hat{b}_c$. Fig. 5.7 provides a visual representation of the internal structure on a ConvLstm cell.

Additionally, in our model, the ConvLstm layer is also used as the initial filter applied to video input before the 3D Sequential DenseConvLstm network. The ConvLstm is initially used to set the number of frames to capture spatiotemporal features. In the proposed model, the number of frames= 10.

- **Average Temporal Pooling:** Even though the ConvLstm efficiently extracts spatiotemporal features but it might be biased towards the end frames in the sequence which can reduce the efficiency in extracting appropriate information over the full sequence. Therefore, we utilize the *temporal average pooling* to capture long-term features present by considering information through the sequence. Additionally, it downsamples the feature map size by setting the stride= 2.

Growth Rate

Each DB produces a feature map of size f (i.e. generated by eq. (5.11)) that is controlled by the growth rate (G). The G is a small integer value that regulates the amount of new information held by each layer. Moreover, it controls the growth of the network and improves parameter efficiency. Therefore, the size of L^{th} layer is $f * (l-1) + f_0$ where f_0 represent the size of initial filter.

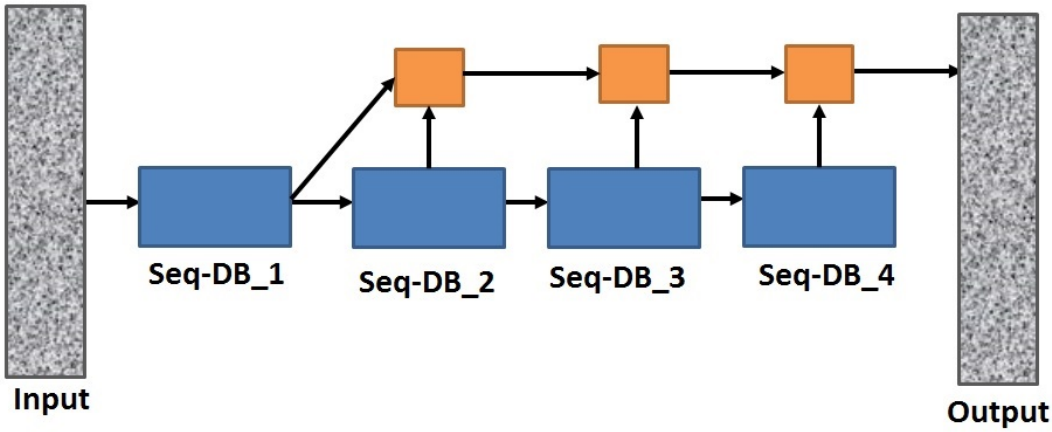


Figure 5.8: Example of Deep Aggregation Process for CNN features from Seq-DB blocks

IDA

Finally, as shown in Fig. 5.5, we aggregate the features extracted from each **Seq-DB**$_N$ to produce the final feature map through *iterative deep aggregation* (F. Yu et al., 2018). The process of aggregation starts with the shallow layers and then iteratively merges with deeper layers. The aggregation between deep and shallow layers has proved in the literature to improve the overall performance of the network with a high-resolution feature map (Xu et al., 2019). The process of aggregation starts with the first shallow layer and keep merging with deeper layers iteratively as follows:

$$F(d_1, ..., d_n) = \begin{cases} d_1 & if \quad n = 1 \\ F(A(d_1, d_2), ..., d_n) & otherwise \end{cases} \tag{5.19}$$

where, d_n represent the features extracted from Seq-DB$_N$ for n=number of the Seq-DB blocks. And, A represents the aggregation node and F final feature map used by the next phase for detection. To combine the features from different layers, we downsample the low-level feature through convolution and merge it with the following level features. Fig. 5.8 shows an example of an Iterative deep aggregation process of $n = 4$ Seq-DBs.

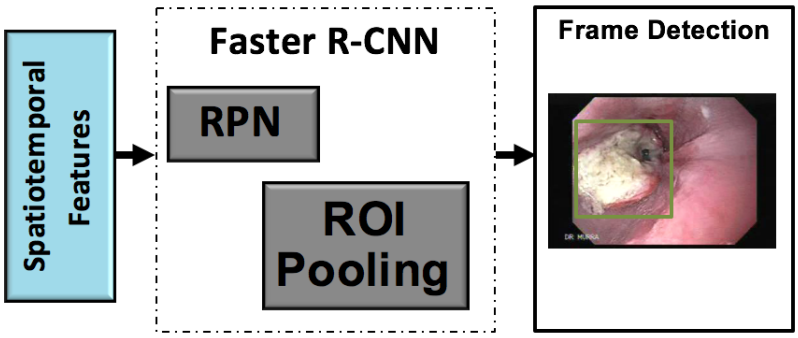


Figure 5.9: Spatiotemporal features are used by the Faster R-CNN to generate bounding boxes output for each frame.

5.3.2 Faster R-CNN

In our model, the feature map produced by the 3D Sequential Dense-ConvLstm is employed by the Faster R-CNN to generate Bounding Box (BB) that detect abnormal esophageal regions in the endoscopic video frames as shown in Fig. 5.9. As a recall, the Faster R-CNN is composed of two main stages: *RPN* and *ROI pooling layer* that share the same feature map to reduce computational complexity. The RPN is responsible for generating a list of candidate BBs with a confidence score that might hold an abnormal region. The RPN depends on anchor boxes that have different sizes and scales to provide N proposals for each location. For each image, there exist $(W \times H \times N)$ proposals, where W and H represent the size of the feature map. Afterward, the ROI pooling unifies the feature map of each proposal generated by the RPN layer and classifies them using softmax into normal, precancerous and cancerous regions. Moreover, the ROI pooling has a regression layer that produces the coordinates of the BB (c_x, c_y, w, h) that locates the detected region. Further details about Faster R-CNN can be found in Chapter 4 (i.e. Section 4.4).

5.3.3 Frame Search Conditional Random Field (FS-CRF)

The target of the post-processing stage is to improve the overall detection performance of the model by removing false positives (FPs) and recovering missing abnormal regions in a sequence of frames obtained from the previous step. The proposed FS-

CRF is constructed on two stages; a *frame search* algorithm and *densely CRF* applied on a frame base level. The pseudocode for the proposed FS-CRF post-processing method is summarized in Algorithm 2. Each stage is described in detail below.

Frame Search algorithm

The *frame search* algorithm has two main functions as shown in Fig. 5.10 : (i) to remove false positives (FPs), and (ii) to recover missing regions, by searching for its nearest labeled frames (i.e. frames with detection).

- *Removing FPs:* For any frame (f) with **detection**, the algorithm searches if there exists another detection within a window threshold t (for example, if $t = 7$ then $+/- 7$ f). If the current frame is the only detection then this frame is counted as FP and the detected BBs are removed.
- *Nearest Labeled Frame (L)*: For any frame (f) with **no detection**, the algorithm search for the nearest frames with detection labels named L_x and L_y within the window frame t. Then it checks if the IoU of the BBs in these frames as follows:

$$IoU = \frac{B_x \cap B_y}{B_x \cup B_y} \tag{5.20}$$

  Where, B_x and B_y represent the area of the generated BB for two nearest frames (L_x, L_y) respectively. If the IoU is greater than 0.7 then these frames are considered to have the same abnormal region. A labeled image (L) is generated from the intersection of the labels of L_x and L_y (as shown in Fig. 6.9c). The label image (L) is then used in the next stage to find missed regions along with frames L_x and L_y.

CRF

Conditional Random Fields (CRF) is a probabilistic graphical method that models complex geometric characteristics such as shape, region context, and information of relations between regions. In our post-processing phase, we adopt the densely CRF

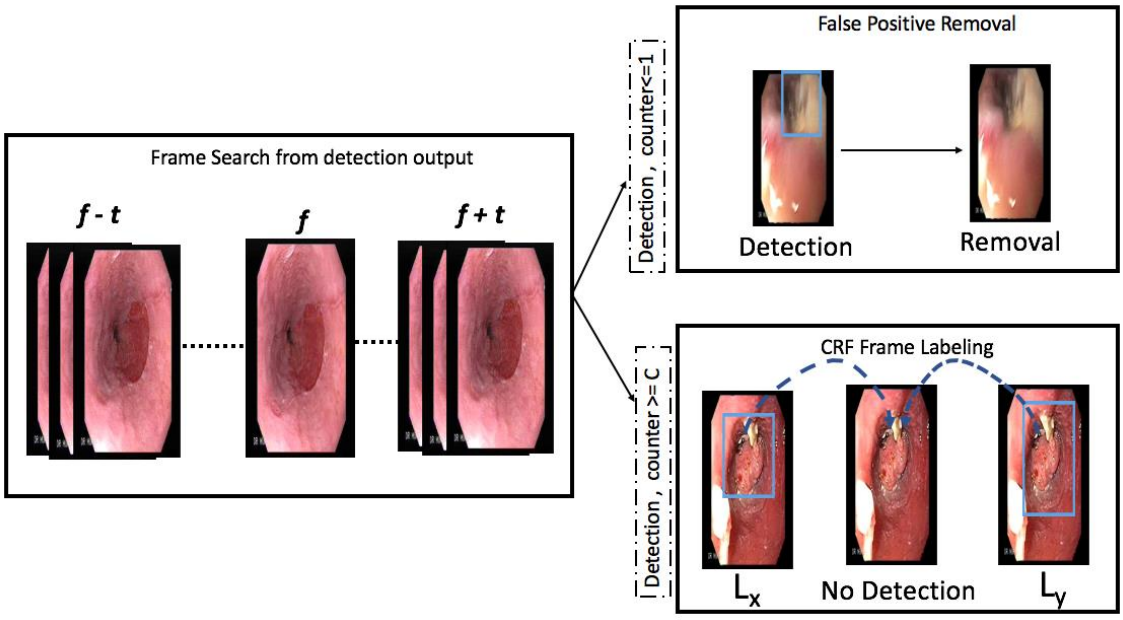


Figure 5.10: The proposed Frame Search algorithm in the post-processing stage has two main functions: (i) To remove False Postive Bounding boxes ash shown in the first row, (ii) to find the two nearest labeled frames to recover regions in missing frames as shown in the second row.

proposed by (Krähenbühl and Koltun, 2011) and apply it on a frame base level using the labeled frame (L) generated from the first stage to find missed abnormal regions in frame (L_f) (i.e. frame with no detection). For an input frame (L_f) and label (L), the CRF energy function is given by:

$$E(l) = \sum_i \sigma_u(l_i) + \sum_{i<j,h} \sigma_p(l_i, l_j, l_h) \tag{5.21}$$

where the unary potential is defined as the negative log-likelihood $\sigma_u(l_i) = -\log Q(l_i|L_f)$ that measure the energy cost of assigning the label (l_i) to pixel i in frame L_f. Where, $Q(l_i|L_f)$ is obtained from the output of Faster R-CNN using the proposed 3D Sequential Dense-ConvLstm for nearest labeled frames (L_x, L_y). The pairwise potential $\sigma_p(l_i, l_j, l_h)$ is defined as a linear combination Gaussian kernels (where i, j and h are pixels from frames L_f, L_x, and L_y respectively) given by:

$$\sigma_p(l_i, l_j, l_h) = \mu(l_i, l_j) \underbrace{\sum_{n=1}^{N} w^n k^n(z_{i,f}, z_{j,x})}_{k(z_{i,f}, z_{j,x})} + \mu(l_i, l_h) \underbrace{\sum_{n=1}^{N} w^n k^n(z_{i,f}, z_{h,y})}_{k(z_{i,f}, z_{h,y})} \quad (5.22)$$

where w^n is the linear combination weight, z_i, z_j and z_h are feature vectors for pixels, i, j and h in an arbitrary feature space, μ represents the label compatibility function and k^n for $n = 1, 2, ..., N$ representing the Gaussian kernels. Following (Krähenbühl and Koltun, 2011), we use **two** kernels defined as:

$$k(z_{i,f}, z_{j,x}) = w_1 \exp(-\frac{|p_{i,f} - p_{j,x}|^2}{2\delta_\alpha^2} - \frac{|I_{i,f} - I_{j,x}|^2}{2\delta_\beta^2}) + w_2 \exp(-\frac{|p_{i,f} - p_{j,x}|^2}{2\delta_\gamma^2}) \quad (5.23)$$

and,

$$k(z_{i,f}, z_{h,y}) = w_1 \exp(-\frac{|p_{i,f} - p_{h,y}|^2}{2\delta_\alpha^2} - \frac{|I_{i,f} - I_{h,y}|^2}{2\delta_\beta^2}) + w_2 \exp(-\frac{|p_{i,f} - p_{h,y}|^2}{2\delta_\gamma^2}) \quad (5.24)$$

The first term in equations (5.23 & 5.24) relies on the pixel position (p) and color (I) (appearance kernel) and the second term depends on (p) only (smoothness kernel). The Gaussian kernel is controlled by the scale (δ).

The target is to minimize the value E(l) of the CRF energy function to produce the most probable label for each pixel in L_f. The energy function is approximately estimated by using the mean-field inference algorithm (Krähenbühl and Koltun, 2011) to compute the distribution $Q(L)$ to create the new label (L). As summarized in Algorithm 3, the approximation of $Q(L)$ is optimized iteratively by applying a sequence of message passing that updates a single variable through incorporating information from the variables of the nearest two frames L_x and L_y. Using the output Label image (CRF_{Label}) from the CRF we generate a bounding box for the unlabeled frame (f) (as shown in Fig. 6.9d and 6.9e).

Fig. 5.11 displays samples of the output after applying the CRF to the unlabelled frame. The figure presents two examples: i) a frame with generated detection and ii) a frame with no detection. This shows that using CRF does not always generate

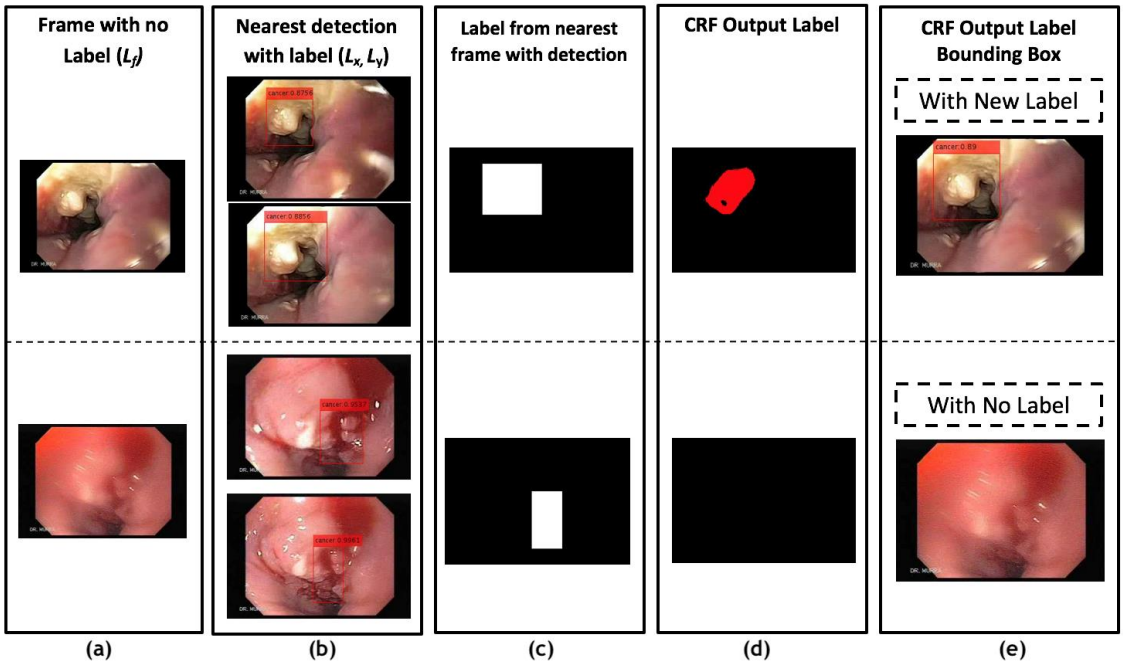


Figure 5.11: Examples of the generated bounding-box using the CRF to find the abnormal region in unlabelled frame L_f using the label from the nearest labeled frames L_x & L_y. The first row displays an example of prediction after FS-CRF post-processing, while the second row represents an example of no prediction.

bounding boxes for each unlabeled frame which proves the robustness of the post-processing phase.

5.4 Experimental Setting and Results

Two experiments were conducted in this section. In the first experiment, the esophageal videos dataset was used for training, validation, and testing of the algorithm. In the second experiment, to assess the robustness of the proposed method, further evaluation using a publicly available colonoscopy video dataset (CVC-DB) was carried to compare our model with video detection results in the literature. In this section, the dataset, implementation, parameter setting of the models, and evaluation protocols are described. Then comprehensive experimental results are presented and discussed in terms of quantitative and qualitative evaluations.

Algorithm 2 Proposed Frame Search Algorithm Steps Description

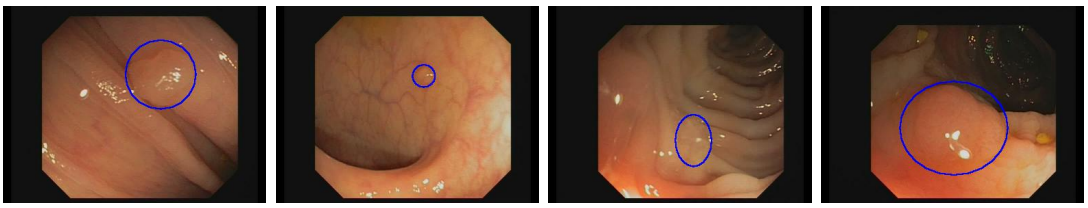
Require: Video frames with BBs from Faster R-CNN using 3D Sequential DenseConvLstm

for $f = 1$ **to** N **do** { N : no. Video frames}
  $counter = 0$
  $Label =$ label for detection of frame f
  if *label* **then** {Has a Label}
    for $i = f + 1$ **to** t **do** {t=frame threshold}
      if *label* **then**
        $counter++$
      end if
    end for
    for $i = f - t$ **to** $f - 1$ **do** {t=frame threshold}
      if *label* **then**
        $counter++$
      end if
    end for
    if $counter <= 1$ **then**
      Remove Label (considered as FP)
    end if
  else {Has **no** Label}
    for $i = f + 1$ **to** t **do** {t=frame threshold}
      if *label* **then**
        $counter++$
        $L_x \leftarrow f$ {frame with nearest label before}
      end if
    end for
    for $i = f - t$ **to** $f - 1$ **do** {t=frame threshold}
      if *label* **then**
        $counter++$
        $L_y \leftarrow f$ {frame with nearest label after}
      end if
    end for
    $IoU_{BB_{xy}} = IoU(BB[L_x], BB[L_y])$
    if $counter >= 2$ **and** $IoU_{BB_{xy}} > 0.7$ **then**
      $L \leftarrow intersection(BB[L_x], BB[L_y]))$
      $L_f \leftarrow$ frame (f) with no label
      $CRF_{Label} \leftarrow CRF(L_f, L_x, L_y, L)$
      Generate BB from CRF_Label for L_f
    end if
  end if
end for

Ensure: Updated video frames BBs from FS-CRF

Algorithm 3 Mean Field Algorithm for Proposed Frame-Based CRF

Require: L_f, L_x, L_y and L
Ensure: Q
1: $Q \leftarrow \frac{1}{Z_i} exp\{-\sigma_u(l_i)\}$
2: **while** not converged **do**
3: $\tilde{Q}_i^{\,n}(l_i) \leftarrow (\sum_{\forall i \neq j} k^n(z_{i,f}, z_{j,x}) Q_j(l_i) + \sum_{\forall i \neq h} k^n(z_{i,f}, z_{h,y}) Q_k(l_i))$
4: $\hat{Q}_i(l_i) \leftarrow \sum_{l \in L} \mu^n(L_{i,f}, l_i) \sum_n w^n \tilde{Q}_i^{\,n}(l_i)$
5: $Q_i(l_i) \leftarrow exp\{-\sigma_u(l_i) - \hat{Q}_i(l_i)\}$
6: normalize $Q_i(l_i)$
7: **end while**



(a) (b) (c) (d)

Figure 5.12: Examples of frames from the CVC-ClinicVideoDB dataset with the annotation of the polyps by the expert in blue.

5.4.1 Dataset

The Gastrointestinal dataset was described in Chapter 2 (Section 2.5.4). The dataset consists of 44 endoscopic videos gathered from 16 patients with different types of esophageal abnormalities (precancerous and cancerous). In the current study, the dataset is split randomly according to patients into 50% training, 20% validation and 30% testing. Samples of frames from the video dataset with the annotations are shown in Chapter 2 (Section 2.5.4).

The colonoscopy dataset "CVC-ClinicVideoDB" (Angermann et al., 2017) is a dataset composed of 18 videos each having a unique polyp appearing in various frames throughout the video. The total number of frames in the 18 videos are 11954 frames, where each polyp are manually annotated by an expert in the field. Since there is no work related to esophageal abnormality detection, we use this dataset to evaluate and compare our model on a similar video dataset used in literature. Fig. 5.12. illustrates different samples from the CVC-ClinicVideoDB with the annotations by the expert.

5.4.2 Implementation Setup

The model is implemented using Keras Libary (Tensorflow backend) on a desktop with Intel Core i7 (3.6GHz processor) and an NVIDIA GeForce GTX1080 Ti with 11GB on a single GPU memory. The weights are initialized randomly with a gaussian distribution ($\mu = 0$, $\sigma = 0.01$). The initial learning rate was set to ($1e-5$) and drops by the factor 0.1 every 1000 iteration and used a weight decay of 0.0004.

To select the parameters of the 3D Sequential Dense-ConvLstm, different values for the *Seq-DB*= 3, 4 and 5 and growth rates (G)= 16, 24 and 32 were evaluated on the dataset as will be shown in the next section. The optimal 3D Sequential Dense-ConvLstm network performance in our model is formed of 5 dense blocks with $G = 24$. Moreover, the initial ConvLstm filter was set to include 10 frames to capture spatiotemporal features. During implementation, we tried to include more number of frames but due to the limited GPU memory, the model could not handle more than 10 frames.

For the *FS-CRF* we tested different window frame t =5, 10, 15, 20 and 25 to find the best performance. As will be discussed, selecting $t = 15$ gave the best results balancing between precision and recall values. Furthermore, we set the IoU=0.7 between the two nearest frames L_x and L_y. We decided to choose a high IoU value to guarantee that the detected region is the same between these two frames. The hyperparameters of the fully-connected CRF were defined in a configuration experiment using a random search on the validation data: $w1 = w2 = 1$, $\delta_\alpha = 80$, $\delta_\beta = 13$ and $\delta_\gamma = 3$. The mean-field algorithm was performed for 10 iterations.

5.4.3 Evaluation Measures

For the esophageal gastrointestinal dataset and the CVC-ClinicVideoDB, the process of automatically detecting the abnormal regions is evaluated using the standard measures *Recall, Precision* and *F-Measure* (explained in Chapter 2 in Section 2.6) to compare with the ground truth annotation. The IoU is used to measure the overlap ratio between the detection results and the manual segmented gold standard which was explained in Chapter 2 (Section (2.6) and Equation (2.6)).

5.4.4 Experimental Results and Discussion

In this section, experiments are carried out to evaluate the performance of the proposed method using the dataset described. We first present the performance of the proposed model with illustrative examples from the detection output. Afterward, we compare the results with and without the proposed FS-CRF post-processing phase. We then compare the model with the corresponding 2D model to evaluate the advantage of processing the video with the 3D model. Later, to justify the design of the 3D Sequential Dense-ConvLstm network, we present a series of experiments to examine the impact of each contribution. Finally, we evaluate our model on colonoscopy video dataset to compare with state-of-the-art results.

Evaluation FS-CRF 3D Seq. Dense-ConvLstm Model

Firstly, we evaluate the performance of our method in detecting the different abnormalities from the endoscopic videos. The results are summarized in Table 5.1 and visualized in Fig. 5.13 and Fig. 5.14. The detection model without the post-processing phase represents a good performance with a recall (88.4%), precision (89.6%) and F-measure (88.9%) which proves the efficiency of the proposed network in extracting relevant spatiotemporal feature from videos. After applying the FS-CRF postprocessing phase to the model, the results have been significantly improved to recall (93.7%), precision (92.7%) and F-measure (93.2%). The proposed FS-CRF attempts to locate abnormal regions missed in intra-frame series caused by any disturbance during movement or in nearby frames. The post-processing boosted the recall performance of the model by 5.3%. Additionally, it effectively removed false positives detected by the network improving the precision by 3.1%.

Moreover, the two tailed T-test was conducted to validate the significance of the detection performance presented in Table 5.1 for the difference between the model with and without the FS-CRF. The test showed that the results were found to be significantly different at the level of 5% (p-value$<$ 0.05).

Additionally, Fig. 5.13 provides different examples of our proposed detection model for the different types of abnormalities (i.e. *BE, EAC, SCC*). Figs 5.13a through

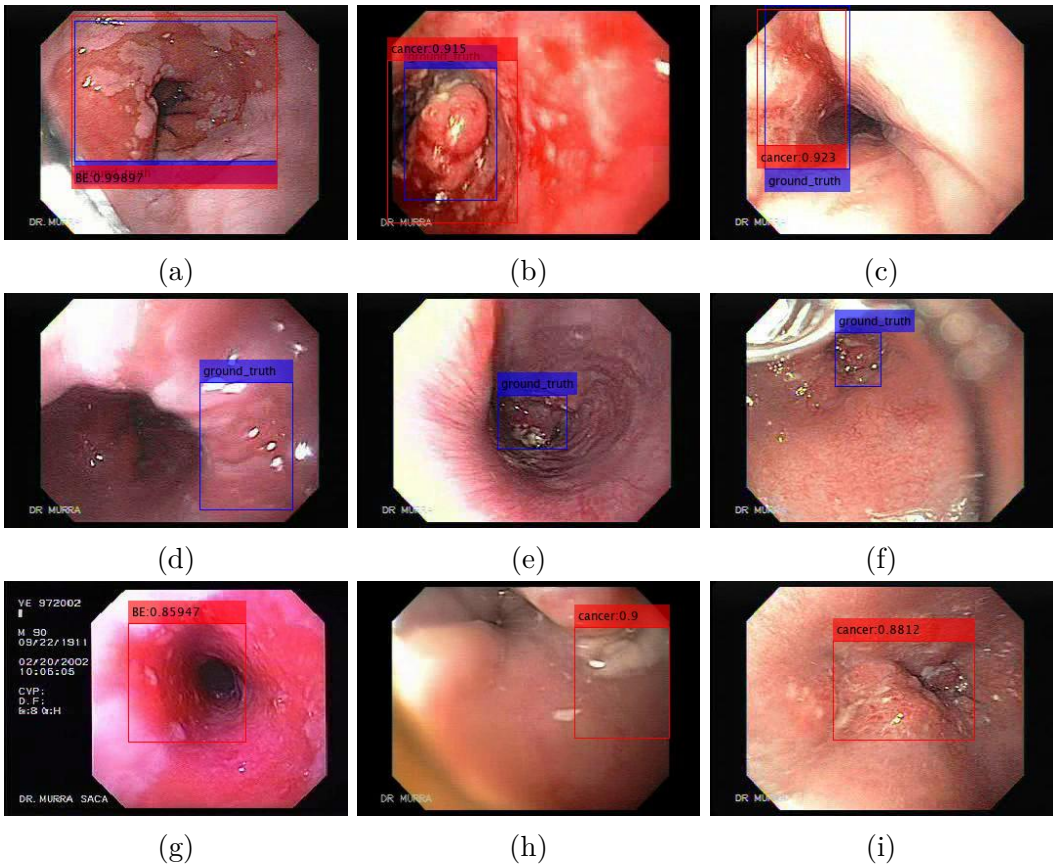
Table 5.1: Detection results of the proposed 3D Sequential DenseConvlStm with and without (w/o) the suggested post-processing FS-CRF methods.

Methods	Recall (%)	Precision (%)	F-Measure (%)
With FS-CRF	**93.7**	**92.7**	**93.2**
W/O FS-CRF	88.4	89.6	88.9

5.13c represent samples of a positive detection for the three abnormal cases, showing the output results in (red bounding box) overlapping with ground truth annotation (purple bounding box). We find that our model can successfully detect the different types of abnormalities with a large IoU with the ground-truth and a high confidence score. The proposed model was not able to detect some abnormalities from different frames as shown in Figs 5.13d, 5.13e and 5.13f. After analyzing the missed abnormal regions, we conclude that most of the missed regions have a challenging appearance in the frame with a relatively small abnormal area. Moreover, Fig. 5.13g to 5.13i show examples of false detection by the proposed model for BE, SCC and EAC respectively.

Furthermore, we observed that the model is able to detect abnormalities from challenging frames (i.e. as explained in section 5.1). Fig. 5.14 illustrates several examples from these results. As displayed, Fig. 5.14a has the appearance of an examination tool and Fig. 5.14b has a lot of bubbles around the tumor, the model effectively detected the cancerous region properly compared to the ground truth. On the other hand, Fig. 5.14c and 5.14d has no ground truth annotation by the expert due to the blurry and fog appearance. As shown, the model successfully located these regions which confirms the robustness of the model. Extracting the spatiotemporal features from the video allowed the model to detect abnormal regions even if they appear in blurry or occluded frames.

Moreover, to evaluate the impact of the proposed FS-CRF post-processing on the model, we calculate the recall and precision values using a varying window frame threshold (t). In Fig. 5.15 we represent the values of the precision and recall at each t =5, 10, 15, 20 and 25 (i.e. t is the number the frames included before and after

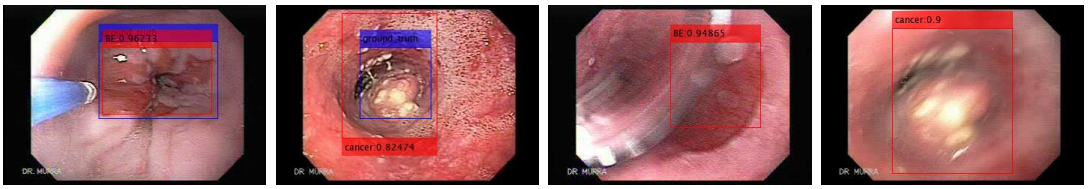


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.13: Examples from the detection output of the proposed FS-CRF 3D Sequential Dense-ConvLstm model. The first-row illustrates samples from positive detection. The second row shows false-negative outputs where the model was not able to locate the abnormality. Finally, the third row represents samples from false positive detection. The three types of abnormalities: *BE*, *SCC*, and *EAC* are represented in First, second and third columns respectively.



(a) (b) (c) (d)

Figure 5.14: Examples of endoscopic challenging frames occluding the esophageal abnormality. (a) Tool appearance, (b) bubbles, (c) blurry, (d) fog.

Table 5.2: Performance comparison between 3D and 2D models without including the FS-CRF post-processing method.

Methods	Recall (%)	Precision (%)	F-Measure (%)
Proposed Model	**88.4**	**89.6**	**88.9**
2D CNN Model	75.8	86.7	80.8

the selected image). Fig. 5.15 demonstrates that the increase of the no. of frames before and after the unlabeled image improves the recall results by detecting more true positives. However, the precision value starts to decrease when including more than $t \geq 20$ frames. The best performance was achieved by the model at $t = 15$ (i.e. results presented in Table 5.1).

Evaluation of 3D Sequential Dense-ConvLstm vs 2D Sequential DenseNet

In this section, we compare the performance of the proposed network (without post-processing phase *(FS-CRF)*) with its 2D version (i.e. 2D Sequential DenseNet) which has the same architecture as the 3D Sequential Dense-ConvLstm but all its layers are 2D instead of 3D and replacing the ConvLstm operation with a 2D Convolutional layer. The reason for this comparison is to investigate the advantage of extracting spatiotemporal features from videos.

As shown, the 2D model achieved a good performance in terms of recall and precision but it still had a notable difference in the performance compared to the 3D model. The 2D obtained a comparable result in terms of the precision value where the 3D model had an increase of only 2.9%. On the other hand, the 3D model outperformed in detecting more abnormal regions increasing the recall value by 12.6%. This result demonstrated the efficiency of the 3D model in dealing with videos to extract spatiotemporal features that improve overall detection performance. Moreover, it can overcome the problem of challenging frames (blurry appearance, tools, bubbles, etc...) while the 2D method failed to detect them.

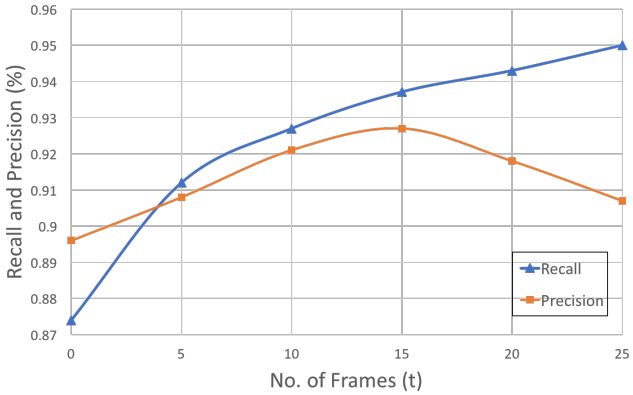


Figure 5.15: The effect of changing the number of frames (t) within the window frame of FS-CRF post-processing on the precision and recall results.

Evaluation of network configuration

Since the 3D-CNN requires a higher computational complexity than the 2D-CNN, therefore, we propose the idea of Sequential DenseNet to simplify the network architecture. The sequential structure can improve computational efficiency while preserving high performance. In Table 5.3, we compare the accuracy (Acc.) performance with the number of trained parameters (Params) for the proposed Sequential DenseNet against Non-Sequential DenseNet. The number of dense blocks = 5 for both networks while varying the growth rate (G) with values: 16, 24 and 32.

As shown, even though the accuracy performance among all the 3 networks is considered comparable for accuracy results, the number of trained parameters is much reduced with the Sequential networks. Therefore, the proposed Sequential DenseConvLstm increased the network's performance with a reduced number of connections and fewer trained parameters. Additionally, the experiments showed that the 3D Seq. DenseConvLstm performs better than the non-sequential network. Increasing the number of layers at later blocks raises the weights of channels holding informative features, reduces the number of layers in earlier blocks and decreases the weights of channels with less beneficial features. The best performance among all networks was achieved for 3D Seq. DenseConvLstm at $G = 24$.

Moreover, Fig. 5.16 represents the AP measure as a function of the IoU threshold for the network with the different configurations. As shown, the different networks

Table 5.3: Performance of 3D Sequential DenseConvLstm and 3D Non-Sequential DenseConvLstm with different growth rate values. The number of Dense Block is fixed as 5 for both networks and growth rate G is selected from three values: 16, 24 and 32. The number of internal layers (l) is set to 5 for the 3D Non-Sequential DenseConvLstm.

Method	**Params** (10^7)	**Acc.** (%)
3D Seq. DenseConvLstm (G=16)	**8.71**	89.15
3D Seq. DenseConvLstm (G=24)	12.01	**91.10**
3D Seq. DenseConvLstm (G=32)	15.56	90.18
3D Non-Seq. DenseConvLstm (G=16)	13.71	88.23
3D Non-Seq. DenseConvLstm (G=24)	20.40	89.78
3D Non-Seq. DenseConvLstm (G=32)	27.98	90.03

had a generally good performance in the detection of abnormal regions with the varying IoU threshold. However, the 3D Seq. DenseConvLstm at $G = 24$ was able to maintain the high performance when compared to other networks, therefore, we set our network to this configuration.

Furthermore, the time needed to generate a detection of bounding boxes using our proposed model was measured. The average time took *2.53* seconds per frame. We believe that the detection speed could be improved when using a more powerful GPU.

Comparison with other methods

To further demonstrate the effectiveness of the proposed detection model, we evaluate the model's performance on another publicly available video dataset used for the examination of the colon. The available dataset is named CVC-ClinicVideoDB dataset (Angermann et al., 2017), which is composed of 18 videos where each video contains a distinctive polyp appearing several times within a sequence of frames. These videos have a total number of 11954 frames with 10052 frames having polyps and annotated by experts.

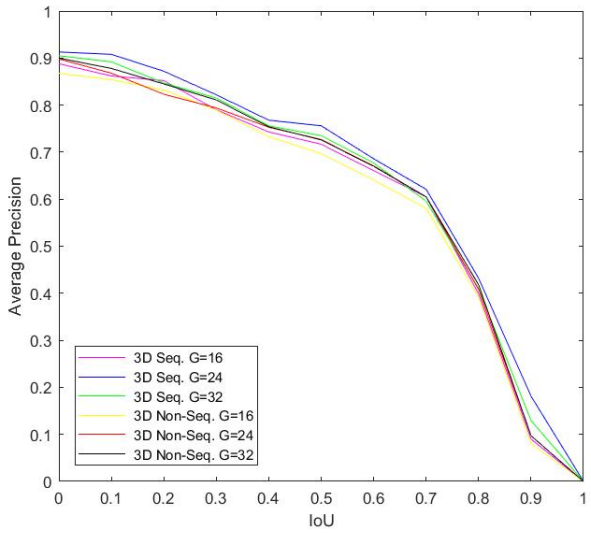


Figure 5.16: AP-IoU threshold curves using different G=16,24 &32 values for 3D Sequential DenseConvLstm and 3D Non-Sequential DenseConvLstm networks.

We compare our results with a recent method suggested in the literature by Qadir et al. (Qadir et al., 2019) that uses the CVC-ClinicVideoDB dataset for evaluation. This model has two phases: First, an object detection method is used to generate region-of-interest (ROI) proposals. Secondly, a False Positive (FP) reduction unit that has a mechanism to detects FPs and correct the outliers of missed polyps in the sequence. The FP unit exploits the temporal dependencies between frames based on the generated region proposals. This method tests two object detection methods separately to generate region proposals: the Faster R-CNN (Ren et al., 2015) with the Inception ResNet (Szegedy, Ioffe et al., 2017) as the CNN backbone network and the Single Shot MultiBox Detector (SSD) (Wei Liu et al., 2016) with the MobileNet (Howard et al., 2017) as the CNN backbone network.

Table 5.4 shows the results of our proposed model when evaluated on the CVC-ClinicVideoDB dataset, where, for our model the dataset is divided randomly according to the full video into 50% training, 10% validation and 40% testing. On the other hand, the method of Qadir et al. trained the model on *selected frames* from colonoscopy videos and evaluated the model on the CVC-ClinicVideoDB dataset. Our results are compared with the model by Qadir et al. when using the Faster

Table 5.4: Comparison of the proposed model results with the method proposed by Qadir et al. (Qadir et al., 2019) using the CVC-ClinicVideoDB dataset (Angermann et al., 2017).

Methods	Recall (%)	Precision (%)	F-Measure (%)
Proposed Model	**81.18**	**96.45**	**88.16**
Faster R-CNN (one ROI) (Qadir et al., 2019)	78.84	90.51	84.27
SSD (one ROI) (Qadir et al., 2019)	53.16	93.03	67.66
Faster R-CNN (five ROIs) (Qadir et al., 2019)	79.75	88.50	83.9
SSD (five ROIs) (Qadir et al., 2019)	53.48	92.57	67.8

R-CNN and the SSD as an object detector model with one and five ROI proposals for the FP reduction unit as suggested by their model.

As shown, the results of our detection model surpassed the results by *Qadir et. al* in all the evaluation measures. By using a more suitable network that can extract spatiotemporal features according to the video properties, our model increased the detection recall by 2.34% and 1.43% when compared to the (Qadir et al., 2019) model using the Faster R-CNN with one and five ROIs respectively. For precision, our method outperformed against all the results obtained by the model is (Qadir et al., 2019) with a value of 96.54% which shows that our model generated much less false positives. In general, the F-measure of our model had the highest performance of 88.16% which indicates the good balance between recall and precision values. Moreover, the proposed post-processing FS-CRF has a fast inference time in generating the updated bounding-box in each frame of the video. On the other hand, in (Qadir et al., 2019) there was a delay in displaying the detection output as the ROI of the current frame depends on the ROIs generated from the surrounding frames.

Moreover, Fig. 5.17 represents various examples of our proposed in detecting polyps from the CVC-ClinicVideoDB dataset. As shown, Figs 5.17a to 5.17c shows successful detection by our model for the polyp with a localized bounding-box around the annotation. Fig 5.17d represents a case where our model was only able to detect one polyp from the frame and missed the other one which was counted as a false negative affecting our recall results. Additionally, Figs. 5.17e & 5.17f shows samples of missed detection where the polyp had a very small structure with a similar view of the surrounding region. Finally, Figs 5.17g to 5.17i represent examples of False positive detection where the model detects similar like areas as polyps. Generally, the model had a good performance in detection polyps from the colonoscopy dataset throughout the evaluation.

5.5 Summary

In this chapter, we present a novel FS-CRF 3D Sequential Dense-ConvLstm model that detects different types of esophageal abnormalities (precancerous and cancerous) from endoscopic videos. The designed features extraction network is capable of learning more representative spatiotemporal features by incorporating the 3D-CNN with the ConvLstm (i.e. covering short and long temporal information), therefore, providing more discriminative features. The proposed network achieved better results when compared to the same 2DCNN network by 8.1% F-measure, which confirms that the proposed network provides more detailed features than features learned only from spatial information. Additionally, the 3D Seq. Dense-ConvLstm layers are constructed in a sequential matter to boost the performance of the network, reduce excessive connection and the number of trained parameters.

Experiments showed that the proposed network had a fewer number of trained parameters with higher efficiency when compared to the non-sequential network. Moreover, we propose a novel post-processing phase that considers information from neighboring frames named FS-CRF to improve the overall performance. To the best of our knowledge, the presented methodology is the first to deal with the detection

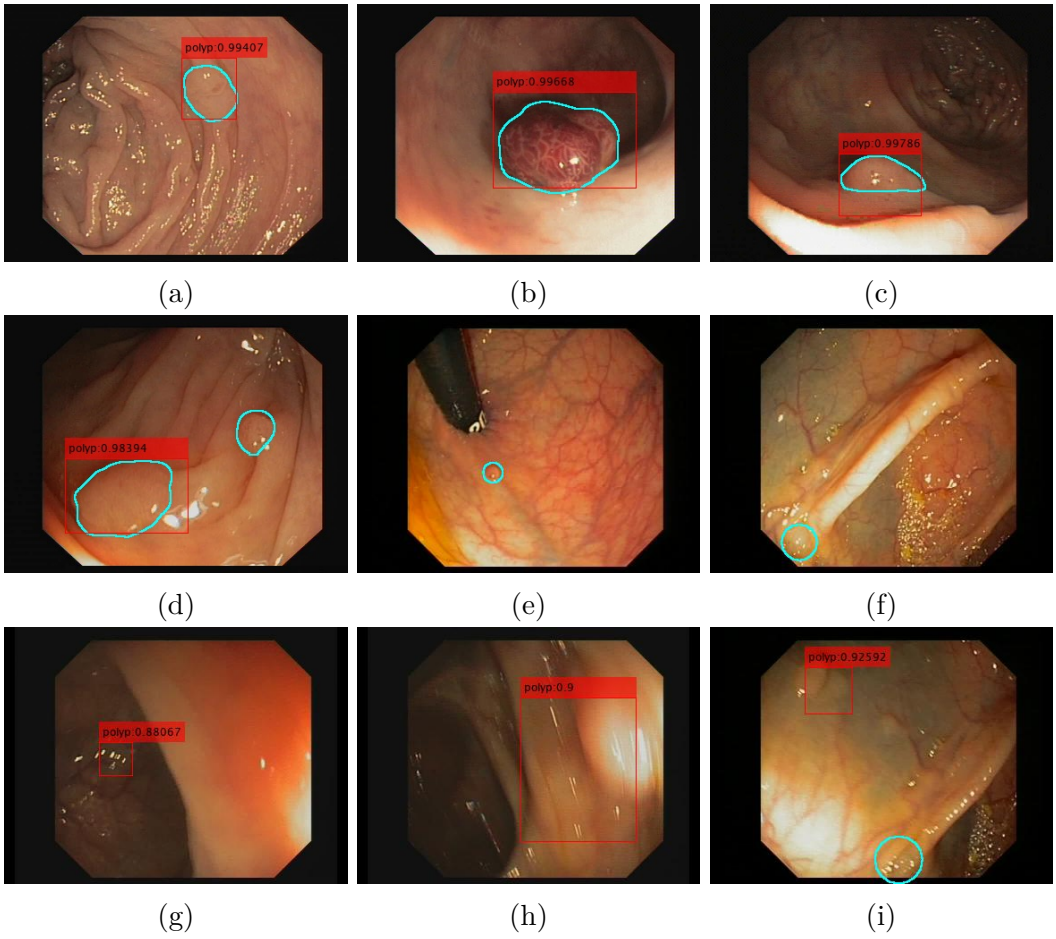


Figure 5.17: Detection examples from the CVC-ClinicVideoDB dataset. The gold-standard by the expert is outlined with blue lines in all the images. The generated bounding box by the model appears in the images with a red bounding box. In the first row, figures (a) to (c) represent correct detection results. In the second row, figure (d) shows an example with two polys where one was detected and the other was missed. Figures (e) & (f) represent samples of false predictions. In the last row. Figures (g), (h) & (i) show a false negative output where the model was not able to predict any abnormality.

of esophageal abnormalities from videos instead of selected frames as discussed in the current litterateurs.

Future research direction includes the investigation of enhancing the model to have a realtime detection to be applied in the clinical routine, segmentation of the abnormal regions and detection of the different artifacts that appear in the endoscopy video during the examination process.

This work has been submitted to the IEEE Journal of Biomedical and Health Informatics (BHI) and currently is under review stage.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

Endoscopy has an essential role in clinical procedures by examining the internal cavities of the patient. Several types of endoscopy in the medical field are available based on the examined organ. The endoscopy is the main tool used for the detection of abnormalities (precancerous and cancerous) appearing in the esophagus tube and the diagnosis is confirmed by taking biopsy samples. Various endoscopic modalities are available for the examination of the esophagus that provide specific information about the abnormal region. The clinical background of the different esophageal abnormality stages and cell deformation were provided in **Chapter 2**. Additionally, the properties and requirements of the various endoscopic modalities were explained. Chapter 2 also described the dataset that was used in this thesis that was publicly available or retrieved from medical imaging conference challenges.

The endoscope provides images/videos for the internal view of the esophagus wall or tissues. Early diagnosis and treatment are important to increase the chances of survival rate. The examination requires a well-experienced physician as the abnormal region has a similar appearance to normal areas and they can be located randomly throughout the tube. Additionally, the physician should be able to provide an instant diagnosis especially with new endoscopic technologies (i.e. such as CLE). Therefore, using computer-aided methods became important and beneficial. In clinical tasks, the results of computer-based automatic methods are used to analyze the image and videos for diagnosis, treatment planning, and prognosis.

In **Chapter 3**, a proposed classification method that used the CLE image as input was investigated to grade the cells into Normal Squamous (NS), Gastric Metaplasia (GM), Intestinal Metaplasia (IM) and Neoplasia (NPL). The main aim of the model is to increase the classification accuracy to assist the doctors as a second opinion before the dysplasia turns into a deadly cancerous, train junior physicians on examining CLE images and help in decreasing the samples that need to be taken through biopsy. The main contribution of this work lies into several aspects:

- A novel enhancement filter is suggested as a preprocessing phase to enhance the internal features of CLE images.
- A multi-scale features are extracted and selected according to the cell deformation properties.
- The proposed method is a single-stage classification model that classifies the type of the image into NS, GM, IM or NPL.
- The method was evaluated on a dataset provided by the ISBI'16 (**AIDA**) challenge and results demonstrated the efficiency of the model in classifying the different grades with high performance.
- The results were compared with the most recent approach in the literature and proved its effectiveness by outperforming state-of-the-art methods on the same dataset achieving a total accuracy 96.05% result of .

In **Chapter 4**, two methods have been developed that successfully detects different esophageal abnormalities from endoscopic images. This chapter was divided into three phases as follows:

- A significant effort has been made to adapt four of the state-of-the-art object detection methods: R-CNN, Fast R-CNN, Faster R-CNN, and SSD to locate abnormal region from the endoscopic images:
  - To the best of our knowledge, this is the first work to evaluate the performance of the deep learning methods with esophageal abnormalities.
  - The evaluation concluded that both the Faster R-CNN and the SSD

provided the best performance when evaluated on two datasets: MICCAI'15 and Kvasir.

  - The SSD had better performance in terms of time complexity but the Faster R-CNN provided more localized bounding boxes around the abnormal regions with higher performance.

- We propose a DenseNet based Faster R-CNN with Gabor Features that uses hybrid features (i.e. handcrafted feature with machine learned features) to detect esophageal abnormal ties from endoscopic images, where:

  - We proposed adopting the DenseNet to extract the CNN features where it can improve the flow of information and the efficiency of parameters throughout the network by reusing learned features from the previous layers.

  - We produce a hybrid feature representation by combining extracted Gabor features with CNN features. Gabor features have shown to provide acceptable performance when employed for the detection of esophageal abnormalities as shown by (Van Der Sommen, F. Zinger S. et al., 2014) and it can distinguish the intestinal juices.

  - Our results demonstrate that the hybrid features (i.e. from the fusion of CNN and Gabour features) have a stronger perception ability than a single image features, therefore, it improved the information used by Faster R-CNN for abnormality detection.

  - The newly designed method was validated on two datasets (Kvasir and MICCAI 2015). Regarding the Kvasir, the results show an outstanding performance with a recall of 90.2% and a precision of 92.1%. While for the MICCAI 2015 dataset, the model showed an exceptional performance with 95% recall and 91% precision.

- A newly designed two input network named GFD Faster R-CNN is proposed to further improved the detection results has been proposed that presented:

  - A generated Gabor Fractal (GF) image that emphasizes the hidden fractal

details of the endoscopic image by maximizing each pixel value based on different Gabor filter responses of the input image.

- The generated GF image is used in a novel two input network model along with the original endoscopic image to detect abnormalities.
- Features are extracted separately from both the GF and endoscopic images using a suggested DenseNet. The extracted features are fused through bilinear fusion before the ROI pooling stage in Faster R-CNN, providing a rich feature representation that boosts the performance of final detection.
- The proposed model surpassed the results of the previous method achieving a recall of 92.7%, precision of 94.2% and F-measure of 93.4% for the Kvasir dataset. While for the MICCAI'15 dataset the proposed model not only improved the results but also outperformed against the state-of-the-art results in the literature with the recall of 97%, precision of 92% and F-measure of 94%.

Chapter 5 represents a novel model to detect abnormalities from endoscopic videos. The process of detection from videos is considered different and more challenging than selected frames (i.e. Images) for several reasons. In the videos, abnormalities can be located anywhere in the frame, partially hidden or covered by other obstacles (i.e. such as intestinal juices, tools, etc...). Moreover, the frames appearing from the endoscopic video can be noisy, blurry, over/under-exposed and with many specular reflections caused by endoscope's light source or movement. In this chapter, we introduce a novel automatic detection model that detects abnormal esophageal regions from endoscopic videos that included :

- A new backbone network named 3D Sequential Dense-ConvLstm is constructed to extract spatiotemporal features from the video to help find abnormal regions especially in challenging frames.
- The network utilizes the 3D-CNN with ConvLstm (i.e. covering short and long term information) to extract information from a sequence of video frames.
- An FS-CRF post-processing model is introduced to improve the overall per-

formance of the model by recovering regions in neighborhood frames within the same clip based on the initial detection output.

- The proposed model was evaluated on the GastroIntestinal Atlas esophagus dataset that covered three types of abnormalities: *BE, EAC* and *SCC*. The extensive evaluation demonstrated the efficiency of the model achieving 93.7% recall, 92.7% precision and 93.2% F-measure.

There is no work available for esophagus abnormality detection from video or using the same data set, accordingly, it was difficult to compare our results with other models. Therefore, to compare with results in the literature, we tested our model on another colonoscopy video dataset for polyp detection. Our model was able to achieve comparable recall results when compared with other methods using the same dataset, nonetheless, our model outperformed in terms of precision which shows that our model has less false positives.

6.2 Future Work

The proposed methods for automatic classification and detection have been evaluated on different available datasets. The evaluation results showed outstanding performances compared to the state-of-the-art results. The potential future directions in technical aspects are summarized below.

- **Deep Learning for grade classification**

  The proposed automatic classification method achieved outstanding accuracy for the classification of the CLE images. However, exhaustive analysis and tests were required to select the suitable handcrafted features to allow a high classification performance. A future direction is looking into other state-of-the-art CNN classification approaches such as DenseNet, VGG'16, AlexNet, and ResNets to investigate machine-learned data-driven features. A large number of feature maps are produced by employing those deep networks and different interpretations can be obtained from these networks for the classification process. Additionally, the studied handcrafted features used in the current

classification model can be included with the new CNN features to generate hybrid features for improved classification.

Exploring deep learning methods for image classification can effectively improve the performance especially after adding more datasets with detailed categories such as Low Grade Dysplasia (LGD) and High Grade Dysplasia (HGD).

- **Segmentation of abnormal regions from endoscopic images**

  Excellent performance has been achieved by the proposed automatic detection model for esophageal abnormalities. In the current work, the model focused on locating abnormal regions by generating bounding boxes around the detected area. In future work, we plan to investigate detection through the segmentation of abnormal regions and the localization of dysplastic lesions. The process of segmentation will provide a more localized detection for the detected region and can help the physician in monitoring the changes/growth of precancerous and cancerous regions. Networks such as Fully Connected Neural Network (FCN) and U-Net can be investigated for the process of segmentation and equipped with our method. Moreover, models such as Mask R-CNN and DeepLab can be studied to find theirs compatibly in detecting and segmenting the abnormal regions.

  Another future direction should be increasing the dataset used for training the model with different types of abnormalities (precancerous and cancerous). In the current model, the datasets used were the MICCAI'15 and the Kvasir which contained one type of abnormality and trained individually.

- **Real-time detection of abnormalities and artifacts from videos**

  The esophageal abnormality detection from videos using the proposed FS-CRF 3D Seq. Dense-ConvLstm model achieved a good accuracy even when tested on a different GI domain (i.e. detecting colon polyps). In the model, the proposed FS-CRF was applied as a postprocessing stage to improve the overall performance based on the standard dense CRF. In the future, we can modify the network to include CRF as RNN so the weights and parameters can be

trained with regular gradient descent. Therefore, all phases of the model can be trained end-to-end.

Additionally, the proposed method can benefit from including the detection of artifacts. In future work, we can gather a dataset that includes artifacts annotation along with the abnormality location. Such detection can help in frame quality assessment and lead to the decision of informative/non-informative frames to reduce the complexity of the video analysis.

Moreover, an important direction that needs to be further studied is reaching a real-time detection process. As the endoscopy is a real-time examination procedure, the physician will benefit from the computer-based system if it capable of detecting regions during the surgery.

Appendix A

List of Publications

- **Journal Publications:**
  - N. Ghatwary, X. Ye, M. Zolgharni, " Esophageal abnormality detection using DenseNet based Faster R-CNN with Gabor features. IEEE Access, 7, pp.84374-84385, 2019.
  - N. Ghatwary, M. Zolgharni, X. Ye, "Early esophageal adenocarcinoma detection using deep learning methods", International journal of computer assisted radiology and surgery (IJCARS), Vol. 14(4), pp.611-621, 2019.
  - N. Ghatwary, A. Ahmed, E. Grisan, H. Jalab, L. Bidaut and X. Ye, "In-vivo Barrett's esophagus digital pathology stage classification through feature enhancement of confocal laser endomicroscopy", Journal of Medical Imaging (JMI), Vol. 6(1), pp 1-12. March 2019.
- **Conference Publications:**
  - N. Ghatwary, M. Zolgharni, X. Ye, "GFD Faster R-CNN: Gabor Fractal DenseNet Faster R-CNN for automatic detection of esophageal abnormalities in endoscopic images", In International Workshop on Machine Learning in Medical Imaging (MLMI), pp. 89-97, 2019.
  - N. Ghatwary, A. Ahmed, X. Ye, "Automated detection of Barrett's esophagus using endoscopic images: a survey", In Medical Imaging and Understanding Analysis (MIUA), pp. 897-908, 2017.
  - N. Ghatwary, A. Ahmed, X. Ye, H. Jalab, "Automatic grade classifica-

tion of Barrett's Esophagus through feature enhancement", In International Society for Optics and Photonics (SPIE), Medical Imaging 2017: Computer-Aided Diagnosis, Vol. 10134, p. 1013433, 2017.

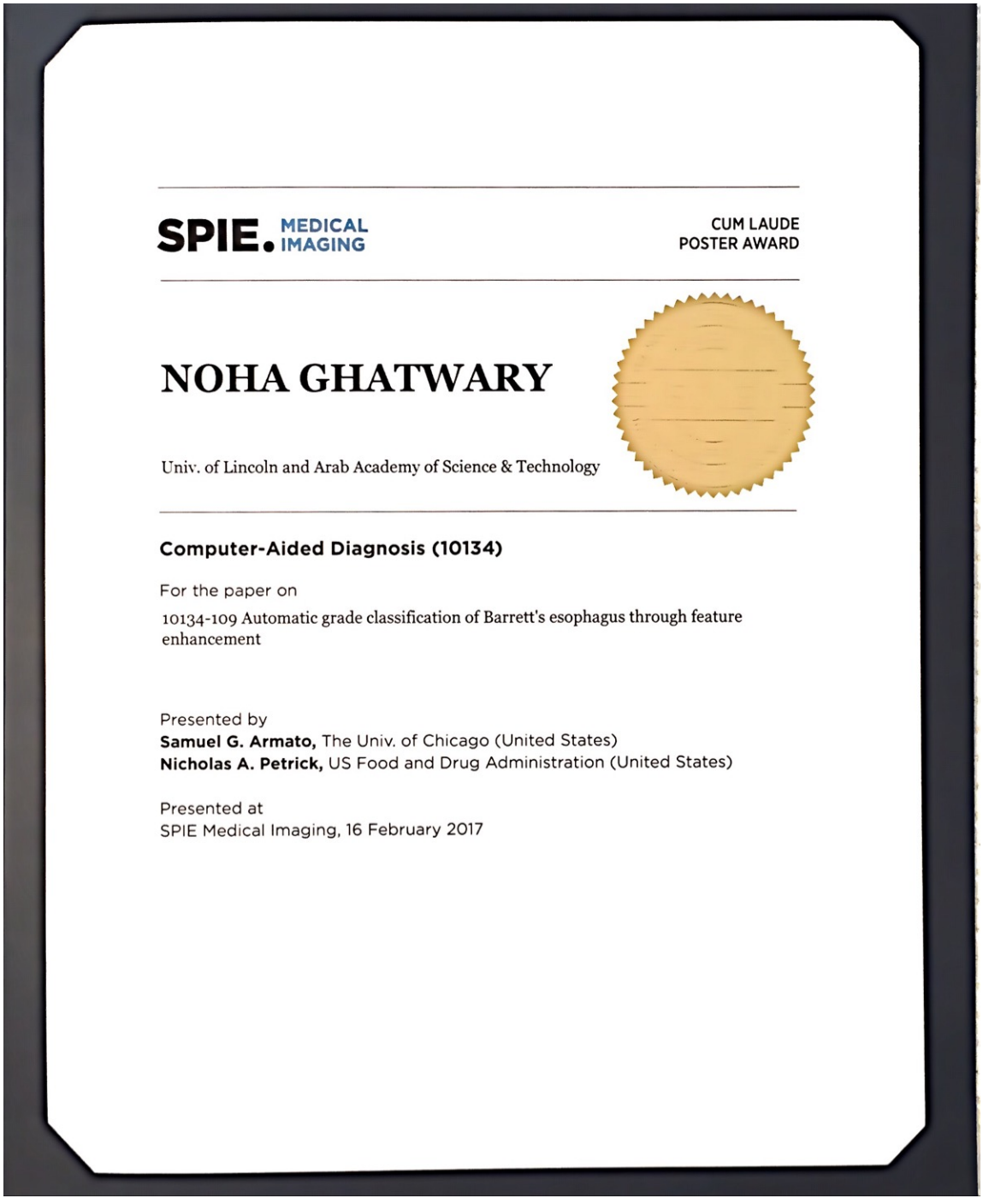
– N. Ghatwary, A. Ahmed, A. and Jalab, H., "Liver CT enhancement using fractional differentiation and integration", In Proceedings of the World Congress on Engineering (WCE), 2016.

– N. Ghatwary, A. Ahmed, A. and Jalab, H., "Liver tumor detection by classification through FD enhancement of CT image", in World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, pp. 2362-2365, 2015.

Appendix B

List of Awards

Figure B.1: Certificate of winning the "Esophagus Micorendoscopy Images in Barrett's Surveillance" challenge



SPIE. MEDICAL IMAGING

CUM LAUDE
POSTER AWARD

NOHA GHATWARY

Univ. of Lincoln and Arab Academy of Science & Technology

Computer-Aided Diagnosis (10134)

For the paper on

10134-109 Automatic grade classification of Barrett's esophagus through feature enhancement

Presented by
Samuel G. Armato, The Univ. of Chicago (United States)
Nicholas A. Petrick, US Food and Drug Administration (United States)

Presented at
SPIE Medical Imaging, 16 February 2017

Figure B.2: Cum Laude award for the best Poster presentation of Computer-Aided Diagnosis

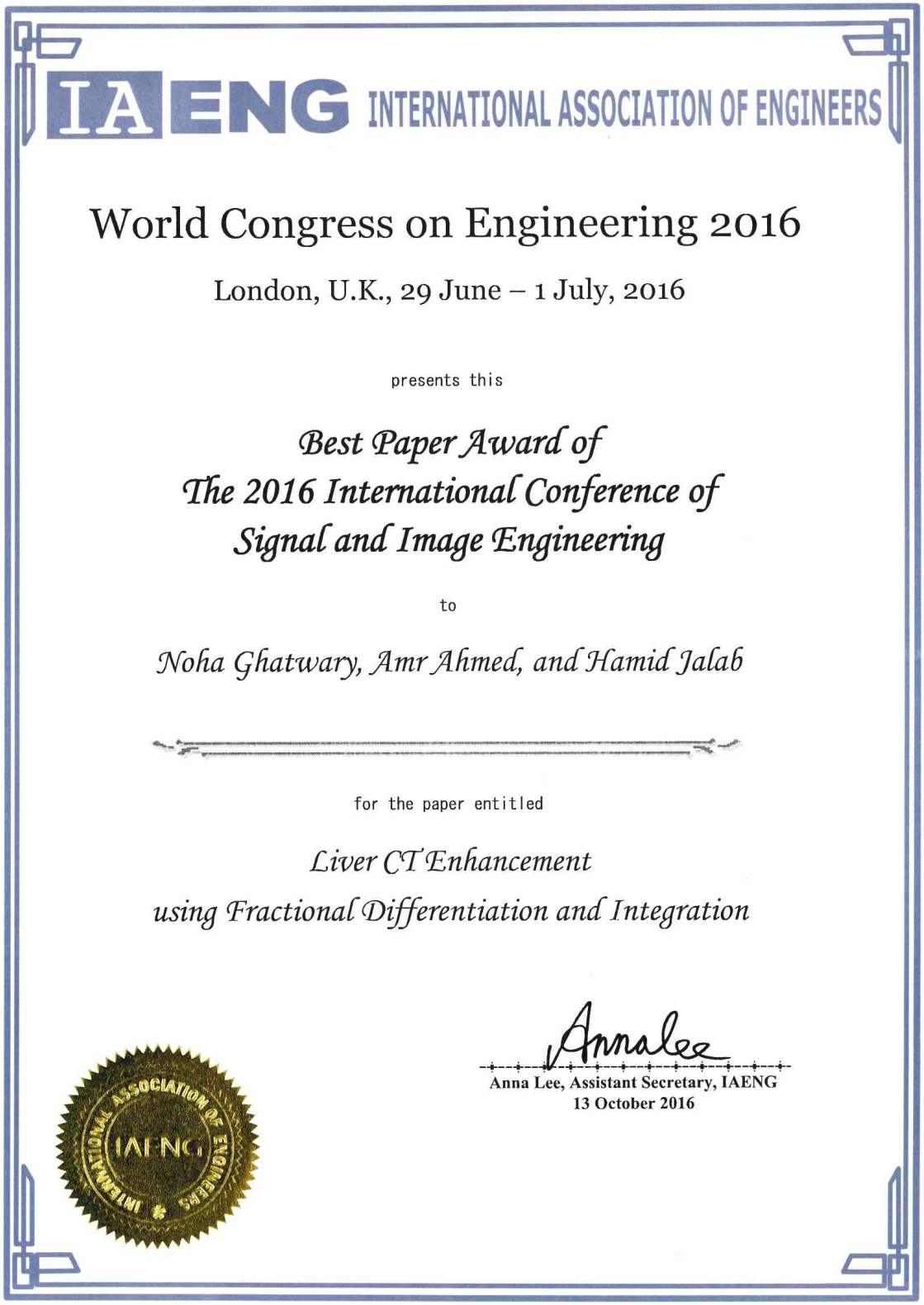


Figure B.3: Best paper award for the paper presented in the WCE conference 2016

Appendix C

Code Samples for Abnormality Grade Classification (Ch. 3)

This appendix includes samples from the Matlab code implemented for the esophageal abnormality grade classification represented in Ch. 3. Moreover, Fig. C.1 represents a graphical abstract for the proposed model.

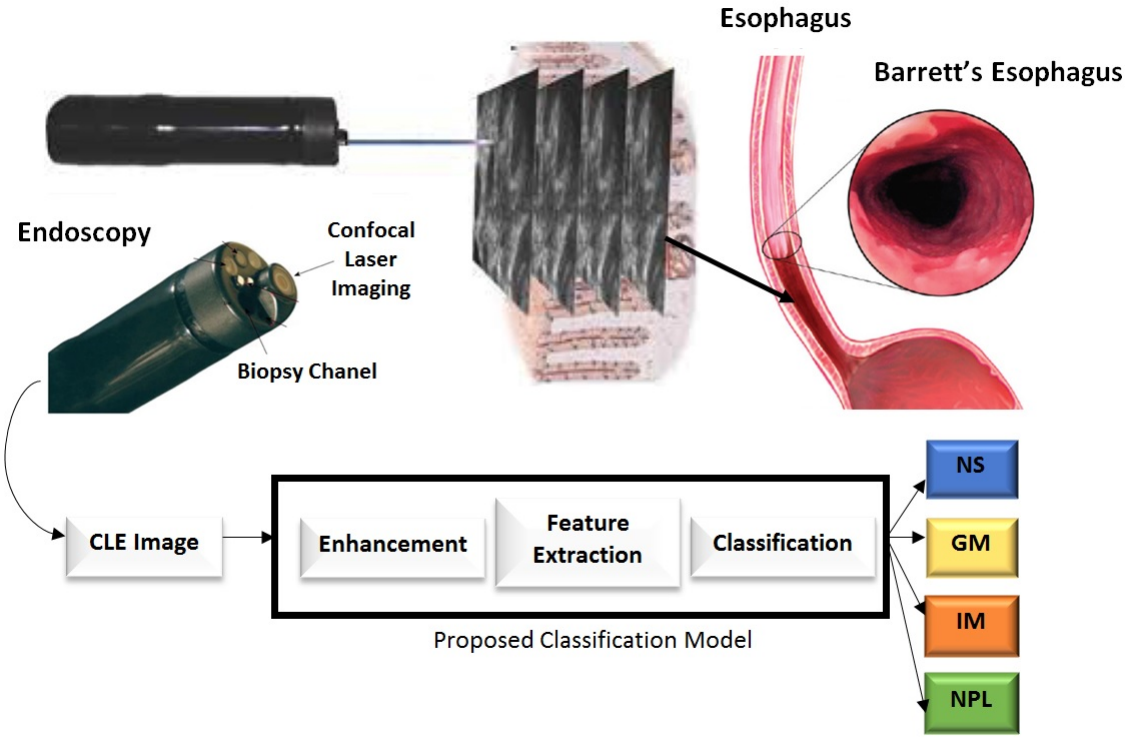


Figure C.1: Graphical abstract for the proposed abnormality pathology grade classification method.

```
%--------------------Grade Classification Code ------------------------
%--------------------- Filter Enhamancect -------------------------------
load('Full_Values.mat')
for i=1:N %N Number of CLE Images from Full Values
lena_I=rgb2gray(imgvalues{i});
[aa,bb,cc,dd]=dwt2(lena_I,'db2');
[bbb]=FI(bb,-0.3);
[ccc]=FI(cc,-0.3);
[ddd]=FD(dd);
A0 = idwt2(aa,bbb,ccc,ddd,'db2');
A1 = uint8(idwt2(aa,bbb,ccc,ddd,'db2'));
fd_2=FD(A1);
img_filter_file{i}=uint8(fd_2);
end
%-------------------------------------------------------------------------
%--------------------- Fractional Diff.----------------------------------
function fd_Value=FD(I,alpha_value)
rand('seed', 672880951);

 for alfa=alpha_value:alpha_value

g=gamma(alfa+1);
% %%%%%%%%%%% Riesz fractional Mask %%%%%%%%%%%%%%%
F0=(2*g)/(gamma(((alfa/2)+1)^2));
F1=(-2*g)/(gamma(alfa/2)*gamma(alfa/2+2));
F2=(2*g)/( gamma((alfa/2)-1))*(gamma((alfa/2)+3) );

%%************Window Mask*****************
Mx=[0,0,0;F0,F1,F2;0,0,0 ];%
hx=Mx;
hy=Mx';
%2-D filtering
X = conv2(I, hx, 'same');
Y = conv2(I, hy, 'same');
gx=abs(X);
gy=abs(Y);
fd_Value=(gx)+(gy);
end
%-------------------------------------------------------------------------
%--------------------- Fractional Intger.---------------------------------

% read original lena image
%lena_I=imread('lena.bmp');
%figure()
%imshow(lena_I)
% add Gauss white noise to lena image
%lena_J=imnoise(lena_I,'gaussian',0,0.01);

% let the intensities of image between 0 and 1
function [image_FI]=FI(Image,v)

I1=Image;
[m,n]=size(I1);
```

```
%v     %The order of fractional differential of digital image.
NEx=4;   % namely mask is NEx*NEx. if choose n=6, the mask we would use is 7*7

% calculate the coefficients of Cs_k
%Cs = zeros(NEx+1);
Cs = zeros(NEx+1);
base=gamma((-v))*((-2*(v))); %the Dominator
Cs(1)=(1)/base;
%Cs(1)=-(steph.^(-v))*0.5*(gamma(v+1))/((gamma(0.5*v+1)^2));


for k=2:NEx
    Cs(k) =(((k+1)^(1-v))-(2*(k^(1-v))) + ((k-1)^(1-v)) )/base;
end
k=k+1;

Cs(k)=(((1-v)*(k^-v)) -(k^(1-v))+((k-1)^(1-v)))/base;


% 8 directions mask image
I11xneg=I1;
I12xpos=I1;
I13yneg=I1;
I14ypos=I1;
I15RDD=I1;
I16LUD=I1;
I17LDD=I1;
I18RUD=I1;

% do convoluting filter
for i=(1+1):1:(m-1)
    for k=(1+1):1:(n-1)
        I11xneg(i,k)=0;
        I12xpos(i,k)=0;
        I13yneg(i,k)=0;
        I14ypos(i,k)=0;
        I15RDD(i,k)=0;
        I16LUD(i,k)=0;
        I17LDD(i,k)=0;
        I18RUD(i,k)=0;
        sumCs1=0;
        sumCs2=0;
        sumCs3=0;
        sumCs4=0;
        sumCs5=0;
        sumCs6=0;
        sumCs7=0;
        sumCs8=0;
        for j=1:1:NEx+1
           if i-j+1>0
           I11xneg(i,k)=I11xneg(i,k)+Cs(j).*I1(i-j+1,k);
           sumCs1=sumCs1+Cs(j);
```

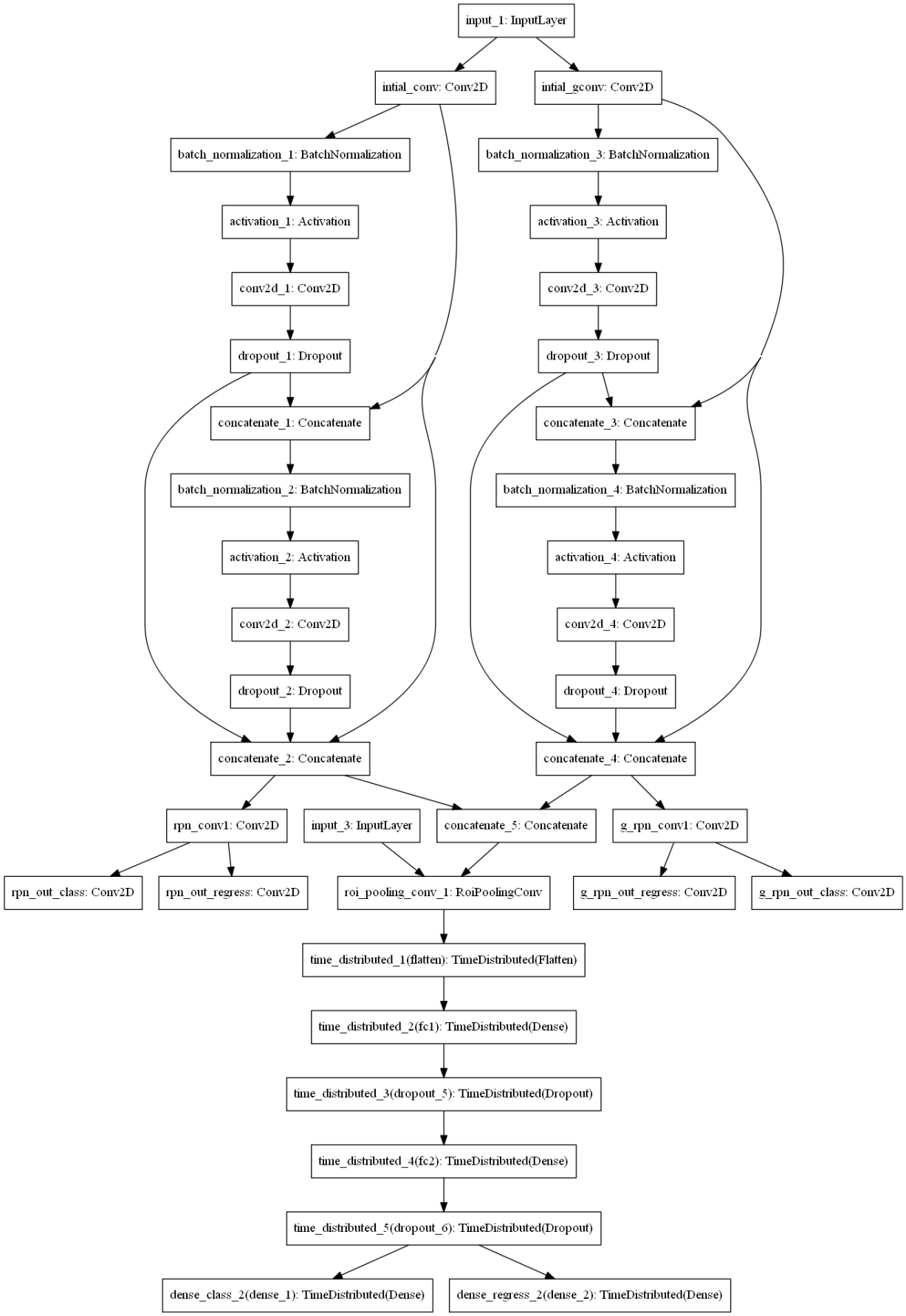
```
            end
            if i+j-1<m+1
            I12xpos(i,k)=I12xpos(i,k)+Cs(j).*I1(i+j-1,k);
            sumCs2=sumCs2+Cs(j);
            end
            if k-j+1>0
            I13yneg(i,k)=I13yneg(i,k)+Cs(j).*I1(i,k-j+1);
            sumCs3=sumCs3+Cs(j);
            end
            if k+j-1<n+1
            I14ypos(i,k)=I14ypos(i,k)+Cs(j).*I1(i,k+j-1);
            sumCs4=sumCs4+Cs(j);
            end
            if (i+j-1<m+1 && k+j-1<n+1)
            I15RDD(i,k)=I15RDD(i,k)+(2.^(-0.5*v))*Cs(j).*I1(i+j-1,k+j-1);
            sumCs5=sumCs5+(2.^(-0.5*v))*Cs(j);
            end
            if (i-j+1>0 && k-j+1>0)
            I16LUD(i,k)=I16LUD(i,k)+(2.^(-0.5*v))*Cs(j).*I1(i-j+1,k-j+1);
            sumCs6=sumCs6+(2.^(-0.5*v))*Cs(j);
            end
            if (i+j-1<m+1 && k-j+1>0)
            I17LDD(i,k)=I17LDD(i,k)+(2.^(-0.5*v))*Cs(j).*I1(i+j-1,k-j+1);
            sumCs7=sumCs7+(2.^(-0.5*v))*Cs(j);
            end
            if (i-j+1>0 && k+j-1<n+1)
            I18RUD(i,k)=I18RUD(i,k)+(2.^(-0.5*v))*Cs(j).*I1(i-j+1,k+j-1);
            sumCs8=sumCs8+(2.^(-0.5*v))*Cs(j);
            end
        end
        newsumCs=sumCs1+sumCs2+sumCs3+sumCs4+sumCs5+sumCs6+sumCs7+sumCs8;
        image_FI(i,k)=(I11xneg(i,k)+I12xpos(i,k)+I13yneg(i,k)+I14ypos(i,k)+I15RDD↙
(i,k)+I16LUD(i,k)+I17LDD(i,k)+I18RUD(i,k))/newsumCs;
    end
end
```

```
%-------------- Feature Extraction from Enhanced
%Images---------------------
load('Filter_FDFI.mat'); % Loading Filtered Images

    for i =1:N

        img = (filter_2{1,i});
        %------------GLCM feature extraction--------
        trainingFeaturesHOG(i,:)=extractHOGFeatures(img,'CellSize',[32 32]);
        glcm=graycomatrix((img));
        glcm_values=graycoprops(glcm);
        trainingFeaturesGLCM(i,:)=[glcm_values.Energy,glcm_values.Homogeneity, ↙
glcm_values.Contrast];
        %------------MP-RLBP-----------------------------------------
        RLBP_I=extractLBPFeatures(img,'Radius',4);
        I1=impyramid(img,'reduce');
        RLBP1=extractLBPFeatures(I1,'Radius',4);
        I2=impyramid(I1,'reduce');
        RLBP2=extractLBPFeatures(I2,'Radius',4);
        I3=impyramid(I2,'reduce');
        RLBP3=extractLBPFeatures(I3,'Radius',4);
        I4=impyramid(I3,'reduce');
        RLBP4=extractLBPFeatures(I4,'Radius',4);
        trainingFeatureslbpPyramid(i,:)=horzcat(RLBP_I,RLBP1,RLBP2,RLBP3,RLBP4);
        %-------MSER------------
       regions = detectMSERFeatures(img);
       [features, valid_points] = extractFeatures(img,regions,'Upright',true);
       trainingFeaturesMSER(i,:)=mean(features);
        %------Fractal Features--------
       trainingFeaturesSFTA(i,:)=sfta(img,4);
      %-------------Fuzzy Local Binary Pattern----------------------------
      trainingFeaturesfLBP(i,:)=flbp(cs);  % fuzzy local binary pattern

    end

   %feature_results_1=horzcat(trainingFeaturesgrey,trainingFeaturesGLCM, ↙
trainingFeaturesSFTA,trainingFeatureslbpPyramid);%,trainingFeaturesHD);
```

Appendix D

Code Samples for Abnormality Detection from Images (Ch. 4)

This appendix includes samples from the python code (i.e. Keras TensorFlow-based) implemented for the esophageal abnormality detection from images represented in Ch. 4. Additionally, a sample of the network created by python for the GFD Faster R-CNN with one dense block and two internal layers is presented.


input_1: InputLayer
intial_conv: Conv2D
intial_gconv: Conv2D
batch_normalization_1: BatchNormalization
batch_normalization_3: BatchNormalization
activation_1: Activation
activation_3: Activation
conv2d_1: Conv2D
conv2d_3: Conv2D
dropout_1: Dropout
dropout_3: Dropout
concatenate_1: Concatenate
concatenate_3: Concatenate
batch_normalization_2: BatchNormalization
batch_normalization_4: BatchNormalization
activation_2: Activation
activation_4: Activation
conv2d_2: Conv2D
conv2d_4: Conv2D
dropout_2: Dropout
dropout_4: Dropout
concatenate_2: Concatenate
concatenate_4: Concatenate
rpn_conv1: Conv2D
input_3: InputLayer
concatenate_5: Concatenate
g_rpn_conv1: Conv2D
rpn_out_class: Conv2D
rpn_out_regress: Conv2D
roi_pooling_conv_1: RoiPoolingConv
g_rpn_out_regress: Conv2D
g_rpn_out_class: Conv2D
time_distributed_1(flatten): TimeDistributed(Flatten)
time_distributed_2(fc1): TimeDistributed(Dense)
time_distributed_3(dropout_5): TimeDistributed(Dropout)
time_distributed_4(fc2): TimeDistributed(Dense)
time_distributed_5(dropout_6): TimeDistributed(Dropout)
dense_class_2(dense_1): TimeDistributed(Dense)
dense_regress_2(dense_2): TimeDistributed(Dense)

```
%-------------DenseNet Network for Faster R-CNN-----------
--------------
%-----------------------------------------------------------
--------------
from __future__ import print_function
from __future__ import absolute_import
from __future__ import division

import warnings

from keras.models import Model
from keras.layers import Flatten, Dense, Input, Conv2D,
MaxPooling2D, Dropout
from keras.layers import GlobalAveragePooling2D,
GlobalMaxPooling2D, TimeDistributed
from keras.engine.topology import get_source_inputs
from keras.utils import layer_utils
from keras.utils.data_utils import get_file
from keras import backend as K
from keras_frcnn.RoiPoolingConv import RoiPoolingConv
from keras.layers.core import Dense, Dropout, Activation
from keras.layers.convolutional import Convolution2D
from keras.layers.pooling import AveragePooling2D
from keras.layers.pooling import GlobalAveragePooling2D
from keras.layers import Input, merge, Concatenate
from keras.layers.normalization import BatchNormalization
from keras.regularizers import l2
from keras.layers import concatenate



def get_img_output_length(width, height):
    def get_output_length(input_length):
        return input_length//16

    return get_output_length(width), get_output_length(
height)


def conv_factory(x, nb_filter, dropout_rate=None,
weight_decay=1E-4):
            """Apply BatchNorm, Relu 3x3Conv2D, optional
dropout

            :param x: Input keras network
```

```
39              :param nb_filter: int -- number of filters
40              :param dropout_rate: int -- dropout rate
41              :param weight_decay: int -- weight decay
    factor
42
43              :returns: keras network with b_norm, relu and
    convolution2d added
44              :rtype: keras network
45              """
46              x = BatchNormalization(mode=0,
47                                     axis=-1,
48                                     gamma_regularizer=l2(
    weight_decay),
49                                     beta_regularizer=l2(
    weight_decay))(x)
50              x = Activation('relu')(x)
51
52
53              x = Conv2D(nb_filter, 3, 3,
54                                 init="he_uniform",
55                                 border_mode="same",
56                                 bias=False,
57                                 W_regularizer=l2(
    weight_decay))(x)
58              if dropout_rate:
59                  x = Dropout(dropout_rate)(x)
60
61              return x
62
63  def transition(x, nb_filter, dropout_rate=None,
    weight_decay=1E-4):
64              """Apply BatchNorm, Relu 1x1Conv2D, optional
    dropout and Maxpooling2D
65
66              :param x: keras model
67              :param nb_filter: int -- number of filters
68              :param dropout_rate: int -- dropout rate
69              :param weight_decay: int -- weight decay
    factor
70
71              :returns: model
72              :rtype: keras model, after applying batch_norm
    , relu-conv, dropout, maxpool
73
74              """
```

```
 75             x = BatchNormalization(mode=0,
 76                                    axis=-1,
 77                                    gamma_regularizer=l2(
    weight_decay),
 78                                    beta_regularizer=l2(
    weight_decay))(x)
 79             x = Activation('relu')(x)
 80
 81
 82             x = Conv2D(nb_filter, 1, 1,
 83                                init="he_uniform",
 84                                border_mode="same",
 85                                bias=False,
 86                                W_regularizer=l2(
    weight_decay))(x)
 87             if dropout_rate:
 88                 x = Dropout(dropout_rate)(x)
 89             x = AveragePooling2D((2, 2), strides=(2, 2))(
    x)
 90
 91             return x
 92
 93 def denseblock(x, nb_layers, nb_filter, growth_rate,
 94                           dropout_rate=None, weight_decay=1E
    -4):
 95             """Build a denseblock where the output of
    each
 96                conv_factory is fed to subsequent ones
 97
 98             :param x: keras model
 99             :param nb_layers: int -- the number of layers
     of conv_
100                               factory to append to the
    model.
101             :param nb_filter: int -- number of filters
102             :param dropout_rate: int -- dropout rate
103             :param weight_decay: int -- weight decay
    factor
104
105             :returns: keras model with nb_layers of
    conv_factory appended
106             :rtype: keras model
107
108             """
109
```

```
110             list_feat = [x]
111 
112 
113             if K.image_dim_ordering() == "th":
114                 concat_axis = 1
115             elif K.image_dim_ordering() == "tf":
116                 concat_axis = -1
117 
118             for i in range(nb_layers):
119                 x = conv_factory(x, growth_rate,
    dropout_rate, weight_decay)
120                 list_feat.append(x)
121                 x = Concatenate()(list_feat)
122                 nb_filter += growth_rate
123 
124             return x, nb_filter
125 #def nn_base(input_tensor=None, trainable=False):
126 
127 def DenseNet(nb_classes, depth, nb_dense_block,
    growth_rate,
128              nb_filter,input_tensor=None,dropout_rate=
    None, weight_decay=1E-4,trainable=False):
129     """ Build the DenseNet model
130 
131     :param nb_classes: int -- number of classes
132     :param img_dim: tuple -- (channels, rows, columns)
133     :param depth: int -- how many layers
134     :param nb_dense_block: int -- number of dense blocks
    to add to end
135     :param growth_rate: int -- number of filters to add
136     :param nb_filter: int -- number of filters
137     :param dropout_rate: float -- dropout rate
138     :param weight_decay: float -- weight decay
139 
140     :returns: keras model with nb_layers of conv_factory
    appended
141     :rtype: keras model
142 
143     """
144 
145     # Determine proper input shape
146     # Determine proper input shape
147     if K.image_dim_ordering() == 'th':
148         input_shape = (3, None, None)
149     else:
```

```
150         input_shape = (None, None, 3)
151 
152     if input_tensor is None:
153         img_input = Input(shape=input_shape)
154     else:
155         if not K.is_keras_tensor(input_tensor):
156             img_input = Input(tensor=input_tensor, shape=
    input_shape)
157         else:
158             img_input = input_tensor
159 
160     if K.image_dim_ordering() == 'tf':
161         bn_axis = 3
162     else:
163         bn_axis = 1
164 
165     model_input = img_input
166     assert (depth - 4) % 3 == 0, "Depth must be 3 N + 4"
167 
168     # layers in each dense block
169     nb_layers = int((depth - 4) / 3)
170 
171     # Initial convolution
172     x = Conv2D(64, (3, 3), activation='relu', padding='
    same', name='intial_conv')(img_input)
173 
174     # Add dense blocks
175     for block_idx in range(nb_dense_block - 1):
176         x, nb_filter = denseblock(x, nb_layers, nb_filter
    , growth_rate,
177                                   dropout_rate=
    dropout_rate,
178                                   weight_decay=
    weight_decay)
179         # add transition
180         x = transition(x, nb_filter, dropout_rate=
    dropout_rate,
181                        weight_decay=weight_decay)
182 
183     x, nb_filter = denseblock(x, nb_layers, nb_filter,
    growth_rate,
184                               dropout_rate=dropout_rate,
185                               weight_decay=weight_decay)
186     #densenet = Model(input=[model_input], output=[x],
    name="DenseNet")
```

```
187
188     return [x,nb_filter]
189
190 def rpn(base_layers, num_anchors,filter_size):
191
192     x = Conv2D(filter_size, (3, 3), padding='same',
    activation='relu', kernel_initializer='normal', name='
    rpn_conv1')(base_layers)
193
194     x_class = Conv2D(num_anchors, (1, 1), activation='
    sigmoid', kernel_initializer='uniform', name='
    rpn_out_class')(x)
195     x_regr = Conv2D(num_anchors * 4, (1, 1), activation='
    linear', kernel_initializer='zero', name='rpn_out_regress
    ')(x)
196
197
198
199     return [x_class, x_regr, base_layers]
200
201
202
```

```
 1 from keras.engine.topology import Layer
 2 import keras.backend as K
 3
 4 if K.backend() == 'tensorflow':
 5     import tensorflow as tf
 6
 7 class RoiPoolingConv(Layer):
 8     '''ROI pooling layer for 2D inputs.
 9     See Spatial Pyramid Pooling in Deep Convolutional
   Networks for Visual Recognition,
10     K. He, X. Zhang, S. Ren, J. Sun
11     # Arguments
12         pool_size: int
13             Size of pooling region to use. pool_size = 7
   will result in a 7x7 region.
14         num_rois: number of regions of interest to be used
15     # Input shape
16         list of two 4D tensors [X_img,X_roi] with shape:
17         X_img:
18         `(1, channels, rows, cols)` if dim_ordering='th'
19         or 4D tensor with shape:
20         `(1, rows, cols, channels)` if dim_ordering='tf'.
21         X_roi:
22         `(1,num_rois,4)` list of rois, with ordering (x,y,
   w,h)
23     # Output shape
24         3D tensor with shape:
25         `(1, num_rois, channels, pool_size, pool_size)`
26     '''
27     def __init__(self, pool_size, num_rois, **kwargs):
28
29         self.dim_ordering = K.image_dim_ordering()
30         assert self.dim_ordering in {'tf', 'th'}, '
   dim_ordering must be in {tf, th}'
31
32         self.pool_size = pool_size
33         self.num_rois = num_rois
34
35         super(RoiPoolingConv, self).__init__(**kwargs)
36
37     def build(self, input_shape):
38         if self.dim_ordering == 'th':
39             self.nb_channels = input_shape[0][1]
40         elif self.dim_ordering == 'tf':
41             self.nb_channels = input_shape[0][3]
```

```
42
43      def compute_output_shape(self, input_shape):
44          if self.dim_ordering == 'th':
45              return None, self.num_rois, self.nb_channels,
    self.pool_size, self.pool_size
46          else:
47              return None, self.num_rois, self.pool_size,
    self.pool_size, self.nb_channels
48
49      def call(self, x, mask=None):
50
51          assert(len(x) == 2)
52
53          img = x[0]
54          rois = x[1]
55
56          input_shape = K.shape(img)
57
58          outputs = []
59
60          for roi_idx in range(self.num_rois):
61
62              x = rois[0, roi_idx, 0]
63              y = rois[0, roi_idx, 1]
64              w = rois[0, roi_idx, 2]
65              h = rois[0, roi_idx, 3]
66
67              row_length = w / float(self.pool_size)
68              col_length = h / float(self.pool_size)
69
70              num_pool_regions = self.pool_size
71
72              #NOTE: the RoiPooling implementation differs
    between theano and tensorflow due to the lack of a resize
    op
73              # in theano. The theano implementation is much
     less efficient and leads to long compile times
74
75              if self.dim_ordering == 'th':
76                  for jy in range(num_pool_regions):
77                      for ix in range(num_pool_regions):
78                          x1 = x + ix * row_length
79                          x2 = x1 + row_length
80                          y1 = y + jy * col_length
81                          y2 = y1 + col_length
```

```
 82
 83                          x1 = K.cast(x1, 'int32')
 84                          x2 = K.cast(x2, 'int32')
 85                          y1 = K.cast(y1, 'int32')
 86                          y2 = K.cast(y2, 'int32')
 87
 88                          x2 = x1 + K.maximum(1,x2-x1)
 89                          y2 = y1 + K.maximum(1,y2-y1)
 90
 91                          new_shape = [input_shape[0],
    input_shape[1],
 92                                       y2 - y1, x2 - x1]
 93
 94                          x_crop = img[:, :, y1:y2, x1:x2]
 95                          xm = K.reshape(x_crop, new_shape)
 96                          pooled_val = K.max(xm, axis=(2, 3
    ))
 97                          outputs.append(pooled_val)
 98
 99              elif self.dim_ordering == 'tf':
100                  x = K.cast(x, 'int32')
101                  y = K.cast(y, 'int32')
102                  w = K.cast(w, 'int32')
103                  h = K.cast(h, 'int32')
104
105                  rs = tf.image.resize_images(img[:, y:y+h,
     x:x+w, :], (self.pool_size, self.pool_size))
106                  outputs.append(rs)
107
108          final_output = K.concatenate(outputs, axis=0)
109          final_output = K.reshape(final_output, (1, self.
    num_rois, self.pool_size, self.pool_size, self.
    nb_channels))
110          print('######################################')
111          xxx = K.print_tensor(final_output, message='My
    softmax values: ')
112          print(xxx)
113
114
115          if self.dim_ordering == 'th':
116              final_output = K.permute_dimensions(
    final_output, (0, 1, 4, 2, 3))
117          else:
118              final_output = K.permute_dimensions(
    final_output, (0, 1, 2, 3, 4))
```

```
119
120         return final_output
121
122
123     def get_config(self):
124         config = {'pool_size': self.pool_size,
125                   'num_rois': self.num_rois}
126         base_config = super(RoiPoolingConv, self).
    get_config()
127         return dict(list(base_config.items()) + list(
    config.items()))
128
```

```
%---------- Gabor Filter Responses--------------
%-----------------------------------------------

import cv2
import numpy as np
import matplotlib.pyplot  as plt

def deginrad(degree):
    radiant = 2*np.pi/360 * degree
    return radiant

############################################################
######################
# cv2.getGaborKernel(ksize, sigma, theta, lambda, gamma,
phi, ktype)
# ksize - size of gabor filter (n, n) --> line width of
road markings in pixel
# sigma - standard deviation of the gaussian function
# theta - orientation of the normal to the parallel
stripes
# lambda - wavelength of the sunusoidal factor
# gamma - spatial aspect ratio
# phi - phase offset
# ktype - type and range of values that each pixel in the
gabor kernel can hold
############################################################
######################
angle=0;
img = cv2.imread(image)

g_kernel=[]
angle=0
col_size = 3
row_size = 3
filter_index = 0
fig, ax = plt.subplots(row_size, col_size, figsize=(12, 8)
)
fig, ax_1 = plt.subplots(row_size, col_size, figsize=(12,
8))
for row in range(0, row_size):
 for col in range(0, col_size):
   theta = deginrad(angle)
   zzz=cv2.getGaborKernel((21, 21), 8.0, theta,  np.cos(
theta), 0.5, 0, ktype=cv2.CV_64F)
   h, w = zzz.shape[:2]
```

```
37    #ax[row][col].imshow(cv2.resize(zzz, (3*w, 3*h),
   interpolation=cv2.INTER_CUBIC))
38    g_kernel=cv2.getGaborKernel((21, 21), 8.0, theta, np.pi
   /4, 0, 0, ktype=cv2.CV_32F)
39    #g_kernel=cv2.getGaborKernel((31, 31), 10, theta, 7, 0.
   5, np.pi, ktype=cv2.CV_32F)
40    filtered_img = cv2.filter2D(img, cv2.CV_8UC3, g_kernel)
41    print('the size of image after',filtered_img.shape)
42    ax[row][col].imshow(filtered_img)
43    ax_1[row][col].imshow(zzz,cmap='Greys')
44    angle = angle + 45
45
46 plt.show()
47
48
49 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
50 filtered_img = cv2.filter2D(img, cv2.CV_8UC3,g_kernel )
51
52 #h, w = g_kernel.shape[:2]
53 #g_kernel = cv2.resize(g_kernel, (3*w, 3*h), interpolation
   =cv2.INTER_CUBIC)
54 #cv2.imshow('gabor kernel (resized)', g_kernel)
55 #cv2.waitKey(0)
56 #cv2.destroyAllWindows()
57 #cv2.waitKey(1)
58 #cv2.imwrite("reference_gabor.png", filtered_img)
59 #cv2.waitKey(0)
60 #cv2.destroyAllWindows()
61 #cv2.waitKey(1)
62
```

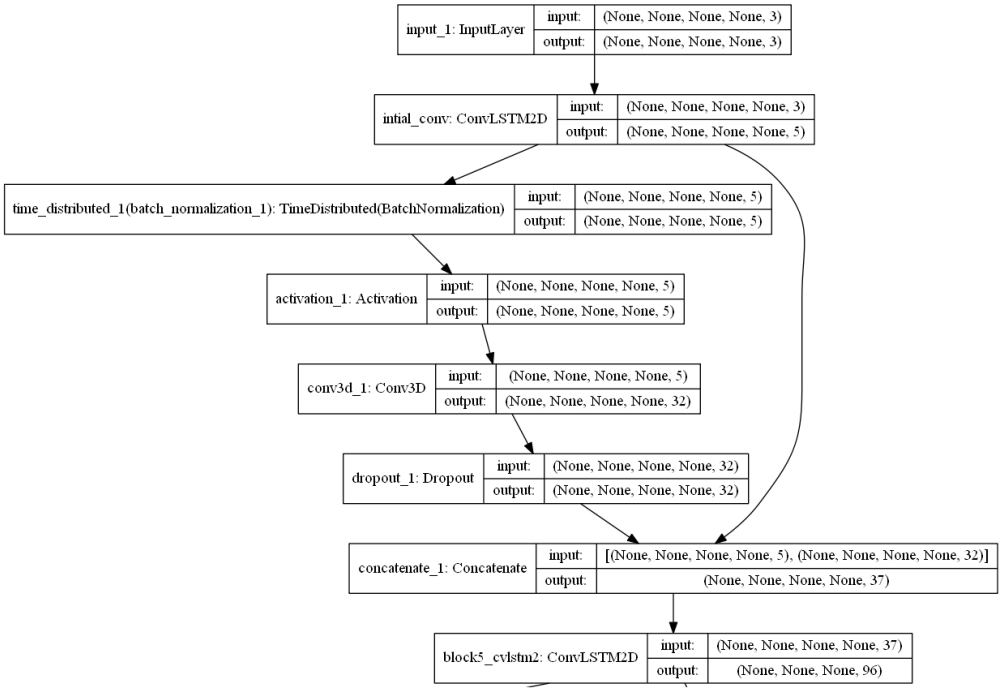
```
%---------- Gabor Fractal Image Generation ------
%------------------------------------------------

from multiprocessing.pool import ThreadPool

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import cv
import cv2

def build_filters(ksize, sigma, a, b, c):
    filters = []
    for theta in np.arange(0, np.pi, np.pi / 16):
        kern = cv2.getGaborKernel((ksize, ksize), sigma,
theta, a, b, c, ktype=cv2.CV_32F)
        kern /= 1.5*kern.sum()
        filters.append(kern)
    return filters

def display_filter (ksize, sigma, a, b, c, fi):
    filters = build_filters (ksize, sigma, a, b, c)
    plt.imshow (filters[fi], cmap = cm.Greys_r)
    plt.show()


def build_gfilters(ksize, sigma):
    filters = []
    kern = cv2.getGaussianKernel(ksize, sigma, ktype=cv2.
CV_32F)
    k2 = kern * cv2.transpose (kern)
    filters.append(kern)
    return filters



def process_threaded(img, filters, threadn = 8):

    def f(kern):
        return cv2.matchTemplate(img, kern, cv.
CV_TM_CCORR_NORMED)
    pool = ThreadPool(processes=threadn)
    accum = None
    for fimg in pool.imap_unordered(f, filters):
        if (accum == None) :
```

```
43              accum = np.zeros_like (fimg)
44          accum += fimg * fimg
45      return accum
46
47  if __name__ == '__main__':
48      import sys
49      from common import Timer
50
51      print __doc__
52      try: img_fn = sys.argv[1]
53      except: img_fn = '../endoscopic_images/.....jpg'
54
55      gimg = cv2.imread(img_fn, 0)
56      img = gimg.astype (np.float32) / 255.0
57      filters = build_filters(127, 15.0, 127.0, 1.0, 0.5)
58      gfilters = build_gfilters (127, 31.0)
59      with Timer('running multi-threaded'):
60          res2 = process_threaded (img, gfilters)
61
62      plt.imshow(res2, cm.Greys_r)
63      plt.show ()
64
65      cv2.imshow('img', img)
66      cv2.waitKey()
67      cv2.destroyAllWindows()
```

Appendix E

Code Samples for Abnormality Detection from Videos (Ch. 5)

This appendix includes samples from the python code (i.e. Keras TensorFlow-based) implemented for the esophageal abnormality detection from videos represented in Ch. 5. Additionally, a sample of the network created by python for the 3D Seq. Dense-ConvLstm with one dense blocks is shown.



```
%---------- 3D Seq. Dense-ConvLstm Network ------
%------------------------------------------------
from __future__ import print_function
from __future__ import absolute_import
from __future__ import division

import warnings

from keras.models import Model
from keras.layers import Flatten, Dense, Input, Conv2D, MaxPooling2D, Dropout,Conv3D
from keras.layers import GlobalAveragePooling2D, GlobalMaxPooling2D, TimeDistributed,ConvLSTM2D, SpatialDropout3D,Lambda,Reshape
from keras.engine.topology import get_source_inputs
from keras.utils import layer_utils
from keras.utils.data_utils import get_file
from keras import backend as K
from keras_frcnn.RoiPoolingConv import RoiPoolingConv
from keras.layers.core import Dense, Dropout, Activation
from keras.layers.convolutional import Convolution2D
from keras.layers.pooling import AveragePooling2D
from keras.layers.pooling import GlobalAveragePooling2D
from keras.layers import Input, merge, Concatenate, concatenate
from keras.layers.normalization import BatchNormalization
from keras.regularizers import l2
from keras.layers import concatenate
#from keras_frcnn import temporal_pooling



def get_weight_path():
    if K.image_dim_ordering() == 'th':
        print('pretrained weights not available for VGG with theano backend')
        return
    else:
        return 'vgg16_weights_tf_dim_ordering_tf_kernels.h5'


def get_img_output_length(width, height):
    def get_output_length(input_length):
        return input_length//16
```

```
40
41      return get_output_length(width), get_output_length(
    height)
42
43
44  def conv_factory(x, nb_filter, dropout_rate=None,
    weight_decay=1E-4):
45              """Apply BatchNorm, Relu 3x3Conv2D, optional
    dropout
46
47              :param x: Input keras network
48              :param nb_filter: int -- number of filters
49              :param dropout_rate: int -- dropout rate
50              :param weight_decay: int -- weight decay
    factor
51
52              :returns: keras network with b_norm, relu and
    convolution2d added
53              :rtype: keras network
54              """
55              x = TimeDistributed(BatchNormalization(mode=0,
56                                      axis=-1,
57                                      gamma_regularizer=l2(
    weight_decay),
58                                      beta_regularizer=l2(
    weight_decay)))(x)
59              x = Activation('relu')(x)
60              x = Conv3D(nb_filter, (3,3,3), activation='
    relu', padding='same')(x)
61
62
63
64              if dropout_rate:
65                  x = SpatialDropout3D(dropout_rate)(x)
66
67              return x
68
69  def transition(x, nb_filter, dropout_rate=None,
    weight_decay=1E-4):
70              """Apply BatchNorm, Relu 1x1Conv2D, optional
    dropout and Maxpooling2D
71
72              :param x: keras model
73              :param nb_filter: int -- number of filters
74              :param dropout_rate: int -- dropout rate
```

```
 75             :param weight_decay: int -- weight decay
    factor
 76
 77             :returns: model
 78             :rtype: keras model, after applying
    batch_norm, relu-conv, dropout, maxpool
 79
 80             """
 81             x = TimeDistributed(BatchNormalization(mode=0
    ,
 82                                    axis=-1,
 83                                    gamma_regularizer=l2(
    weight_decay),
 84                                    beta_regularizer=l2(
    weight_decay)))(x)
 85             x = Activation('relu')(x)
 86             x = ConvLSTM2D(filters=nb_filter, kernel_size
    =(1, 1), border_mode='same',
 87                            return_sequences=True,
    W_regularizer=l2(weight_decay))(x)
 88
 89
 90             if dropout_rate:
 91                 x = Dropout(dropout_rate)(x)
 92
 93             x = AveragePooling3D((2,2,2), strides=(2,2,2)
    )(x)
 94
 95             return x
 96
 97 def denseblock(x, nb_layers, nb_filter, growth_rate,
 98                        dropout_rate=None, weight_decay=1E
    -4):
 99             """Build a denseblock where the output of
    each
100                conv_factory is fed to subsequent ones
101
102             :param x: keras model
103             :param nb_layers: int -- the number of layers
     of conv_
104                               factory to append to the
    model.
105             :param nb_filter: int -- number of filters
106             :param dropout_rate: int -- dropout rate
107             :param weight_decay: int -- weight decay
```

```
107 factor
108 
109             :returns: keras model with nb_layers of
    conv_factory appended
110             :rtype: keras model
111 
112             """
113 
114             list_feat = [x]
115 
116 
117             if K.image_dim_ordering() == "th":
118                 concat_axis = 1
119             elif K.image_dim_ordering() == "tf":
120                 concat_axis = -1
121 
122             for i in range(nb_layers):
123                 x = conv_factory(x, growth_rate,
    dropout_rate, weight_decay)
124                 list_feat.append(x)
125                 x = Concatenate()(list_feat)
126                 nb_filter += growth_rate
127 
128             return x, nb_filter
129 #def nn_base(input_tensor=None, trainable=False):
130 
131 def DenseNet(nb_classes, depth, nb_dense_block,
    growth_rate,
132              nb_filter,input_tensor=None,dropout_rate=
    None, weight_decay=1E-4,trainable=False):
133     """ Build the DenseNet model
134 
135     :param nb_classes: int -- number of classes
136     :param img_dim: tuple -- (channels, rows, columns)
137     :param depth: int -- how many layers
138     :param nb_dense_block: int -- number of dense blocks
    to add to end
139     :param growth_rate: int -- number of filters to add
140     :param nb_filter: int -- number of filters
141     :param dropout_rate: float -- dropout rate
142     :param weight_decay: float -- weight decay
143 
144     :returns: keras model with nb_layers of conv_factory
    appended
145     :rtype: keras model
```

```
146
147     """
148
149     # Determine proper input shape
150     # Determine proper input shape
151     if K.image_dim_ordering() == 'th':
152         input_shape = (3, None, None)
153     else:
154         input_shape = (None, None, 3)
155
156     if input_tensor is None:
157         img_input = Input(shape=input_shape)
158     else:
159         if not K.is_keras_tensor(input_tensor):
160             img_input = Input(tensor=input_tensor, shape=
    input_shape)
161         else:
162             img_input = input_tensor
163
164     if K.image_dim_ordering() == 'tf':
165         bn_axis = 3
166     else:
167         bn_axis = 1
168
169     model_input = img_input
170     assert (depth - 4) % 3 == 0, "Depth must be 3 N + 4"
171
172     # layers in each dense block
173     nb_layers = int((depth - 4) / 3)
174     nb_layers=0
175
176     # Initial convolution
177     x = ConvLSTM2D(filters=64, kernel_size=(3, 3),
    border_mode='same', return_sequences=True, name='
    intial_conv')(img_input)
178
179     # Add dense blocks
180     for block_idx in range(nb_dense_block - 1):
181         #print('focussss with layers first',block_idx)
182         nb_layers=nb_layers+1
183         x, nb_filter = denseblock(x, nb_layers, nb_filter
    , growth_rate,
184                                   dropout_rate=
    dropout_rate,
185                                   weight_decay=
```

```
185 weight_decay)
186
187         #### Saving the  each layer feature map of third
    layer#####################
188         if block_idx==1:
189             cnn_layer=x
190             cnn_layer= AveragePooling3D((2, 2,2), strides
    =(2,2, 2))(cnn_layer1)
191             shared_layers_new = cnn_layer
192         else:
193             shared_layers_new = concatenate([x,
    shared_layers_new], axis=3)
194             shared_layers_new = AveragePooling3D((2, 2, 2
    ), strides=(2, 2, 2))(shared_layers_new)
195
196
197
198
199
200         # add transition
201         x = transition(x, nb_filter, dropout_rate=
    dropout_rate,
202                        weight_decay=weight_decay)
203
204     # The last denseblock does not have a transition
205     nb_layers=nb_layers+1
206
207     x, nb_filter = denseblock(x, nb_layers, nb_filter,
    growth_rate,
208                               dropout_rate=dropout_rate,
209                               weight_decay=weight_decay)
210     x = ConvLSTM2D(nb_filter, kernel_size=(3, 3),
    border_mode='same', name='block5_cvlstm2')(x)
211
212     totallayers=totallayers+nb_filter
213
214     return [shared_layers_new,totallayers]
215
216 def rpn(base_layers, num_anchors,filter_size):
217
218     x = Conv2D(filter_size, (3, 3), padding='same',
    activation='relu', kernel_initializer='normal', name='
    rpn_conv1')(base_layers)
219
220     x_class = Conv2D(num_anchors, (1, 1), activation='
```

```
220 sigmoid', kernel_initializer='uniform', name='
    rpn_out_class')(x)
221     x_regr = Conv2D(num_anchors * 4, (1, 1), activation='
    linear', kernel_initializer='zero', name='rpn_out_regress
    ')(x)
222
223
224
225     return [x_class, x_regr, base_layers]
226
227
228 def classifier(base_layers, input_rois, num_rois,
    filter_size,nb_classes, trainable=False):
229
230     # compile times on theano tend to be very high, so we
     use smaller ROI pooling regions to workaround
231
232     if K.backend() == 'tensorflow':
233         pooling_regions = 7
234         input_shape = (num_rois,7,7,filter_size)
235     elif K.backend() == 'theano':
236         pooling_regions = 7
237         input_shape = (num_rois,filter_size,7,7)
238
239     out_roi_pool = RoiPoolingConv(pooling_regions,
    num_rois)([base_layers, input_rois])
240
241     out = TimeDistributed(Flatten(name='flatten'))(
    out_roi_pool)
242     out = TimeDistributed(Dense(4096, activation='relu',
    name='fc1'))(out)
243     out = TimeDistributed(Dropout(0.5))(out)
244     out = TimeDistributed(Dense(4096, activation='relu',
    name='fc2'))(out)
245     out = TimeDistributed(Dropout(0.5))(out)
246
247     out_class = TimeDistributed(Dense(nb_classes,
    activation='softmax', kernel_initializer='zero'), name='
    dense_class_{}'.format(nb_classes))(out)
248     # note: no regression target for bg class
249     out_regr = TimeDistributed(Dense(4 * (nb_classes-1),
    activation='linear', kernel_initializer='zero'), name='
    dense_regress_{}'.format(nb_classes))(out)
250
251     return [out_class, out_regr]
```

References

Achanta, Radhakrishna et al. (2012). 'SLIC superpixels compared to state-of-the-art superpixel methods'. In: *IEEE transactions on pattern analysis and machine intelligence* 34.11, pp. 2274–2282 (cit. on p. 78).

Adelson, Edward H et al. (1984). 'Pyramid methods in image processing'. In: *RCA engineer* 29.6, pp. 33–41 (cit. on p. 48).

aidasub-clebarrett - Home (Nov. 2015). URL: `https://aidasub-clebarrett.grand-challenge.org` (cit. on pp. 6, 29, 57, 59).

Akilan, Thangarajah et al. (2019). 'A 3D CNN-LSTM-Based Image-to-Image Foreground Segmentation'. In: *IEEE Transactions on Intelligent Transportation Systems* (cit. on p. 140).

Almeida, Ricardo, Dina Tavares and Delfim FM Torres (2019). *The variable-order fractional calculus of variations*. Springer (cit. on p. 42).

Angermann, Quentin et al. (2017). 'Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis'. In: *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures*. Springer, pp. 29–41 (cit. on pp. 150, 157, 159).

Bay, Herbert, Tinne Tuytelaars and Luc Van Gool (2006). 'Surf: Speeded up robust features'. In: *European conference on computer vision*. Springer, pp. 404–417 (cit. on p. 77).

Becker, V et al. (2008). 'Confocal laser scanning fluorescence microscopy for in vivo determination of microvessel density in Barrett's esophagus.' In: *Endoscopy* 40.11, pp. 888–891 (cit. on p. 18).

Beg, Sabina, Ana Wilson and Krish Ragunath (2016). 'The use of optical imaging techniques in the gastrointestinal tract'. In: *Frontline gastroenterology* 7.3, pp. 207–215 (cit. on p. 18).

Behrens, Angelika et al. (2011). 'Barrett's adenocarcinoma of the esophagus: better outcomes through new methods of diagnosis and treatment'. In: *Deutsches Ärzteblatt International* 108.18, p. 313 (cit. on p. 17).

Bird-Lieberman, EL and RC Fitzgerald (2009). 'Early diagnosis of oesophageal cancer'. In: *British journal of cancer* 101.1, pp. 1–6 (cit. on p. 3).

Boschetto, Davide, Gloria Gambaretto and Enrico Grisan (2016). 'Automatic classification of endoscopic images for premalignant conditions of the esophagus'. In: *SPIE Medical Imaging*. International Society for Optics and Photonics, pp. 978808–978808 (cit. on p. 78).

Brooks, Philip J et al. (2009). 'The alcohol flushing response: an unrecognized risk factor for esophageal cancer from alcohol consumption'. In: *PLoS medicine* 6.3 (cit. on p. 1).

Buchner, Anna M and Michael B Wallace (2015). 'In-vivo microscopy in the diagnosis of intestinal neoplasia and inflammatory conditions'. In: *Histopathology* 66.1, pp. 137–146 (cit. on p. 18).

Cai, Zhaowei and Nuno Vasconcelos (2018). 'Cascade r-cnn: Delving into high quality object detection'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162 (cit. on p. 25).

Can the lower esophageal sphincter be observed? (N.d.). URL: `https://www.refluxmd.com/can-lower-esophageal-sphincter-observed/` (cit. on p. 13).

Cancer Stat Facts: Esophageal Cancer (n.d.). URL: `https://seer.cancer.gov/statfacts/html/esoph.html` (cit. on p. 1).

Cao, Zhantao et al. (2017). 'Breast tumor detection in ultrasound images using deep learning'. In: *International Workshop on Patch-based Techniques in Medical Imaging*. Springer, pp. 121–128 (cit. on p. 87).

Castellano, G et al. (2004). 'Texture analysis of medical images'. In: *Clinical radiology* 59.12, pp. 1061–1069 (cit. on p. 43).

Cequera, A and MC Garcia de Leon Mendez (2014). 'Biomarkers for liver fibrosis: advances, advantages and disadvantages'. In: *Revista de Gastroenterología de México (English Edition)* 79.3, pp. 187–199 (cit. on p. 5).

Chao, Wei-Lun, Hanisha Manickavasagan and Somashekar G Krishna (2019). 'Application of artificial intelligence in the detection and differentiation of colon polyps: a technical review for physicians'. In: *Diagnostics* 9.3, p. 99 (cit. on p. 131).

Chen, Wei-Ta, Wei-Chuan Liu and Ming-Syan Chen (2010). 'Adaptive color feature extraction based on image color distributions'. In: *IEEE Transactions on image processing* 19.8, pp. 2005–2016 (cit. on p. 32).

Chen, Yushi et al. (2017). 'Hyperspectral images classification with Gabor filtering and convolutional neural network'. In: *IEEE Geoscience and Remote Sensing Letters* 14.12, pp. 2355–2359 (cit. on p. 67).

Cho, Jin Woong (2013). 'The role of endoscopic ultrasonography in T staging: early gastric cancer and esophageal cancer'. In: *Clinical endoscopy* 46.3, p. 239 (cit. on p. 2).

Cho, Kyunghyun et al. (2014). 'Learning phrase representations using RNN encoder-decoder for statistical machine translation'. In: *arXiv preprint arXiv:1406.1078* (cit. on p. 136).

Coleman, Helen G et al. (2014). 'Symptoms and endoscopic features at Barrett's esophagus diagnosis: implications for neoplastic progression risk'. In: *The American journal of gastroenterology* 109.4, pp. 527–534 (cit. on p. 14).

Cortes, Corinna and Vladimir Vapnik (Sept. 1995). 'Support-vector networks'. In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018 (cit. on p. 35).

Costa, Alceu Ferraz, Gabriel Humpire-Mamani and Agma Juci Machado Traina (2012). 'An efficient algorithm for fractal analysis of textures'. In: *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on*. IEEE, pp. 39–46 (cit. on p. 49).

Cross, George R and Anil K Jain (1983). 'Markov random field texture models'. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1, pp. 25–39 (cit. on p. 34).

Daubechies, Ingrid (1992). *Ten lectures on wavelets*. SIAM (cit. on p. 42).

De Palma, Giovanni D (2009). 'Confocal laser endomicroscopy in the "in vivo" histological diagnosis of the gastrointestinal tract'. In: *World journal of gastroenterology: WJG* 15.46, p. 5770 (cit. on p. 18).

De Souza, Luis Antonio et al. (2017). 'Barrett's esophagus identification using optimum-path forest'. In: *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. Ieee, pp. 308–314 (cit. on p. 78).

Dik, Vincent K, Leon MG Moons and Peter D Siersema (2014). 'Endoscopic innovations to increase the adenoma detection rate during colonoscopy'. In: *World journal of gastroenterology: WJG* 20.9, p. 2200 (cit. on p. 2).

Domingues, Inês et al. (2019). 'Computer vision in esophageal cancer: a literature review'. In: *IEEE Access* (cit. on p. 4).

Du, Wenju et al. (2019). 'Review on the Applications of Deep Learning in the Analysis of Gastrointestinal Endoscopy Images'. In: *IEEE Access* 7, pp. 142053–142069 (cit. on p. 131).

East, James E et al. (2016). 'Advanced endoscopic imaging: European Society of Gastrointestinal Endoscopy (ESGE) technology review'. In: *Endoscopy* 48.11, pp. 1029–1045 (cit. on p. 18).

Ebigbo, Alanna, Robert Mendel et al. (2019). 'Computer-aided diagnosis using deep learning in the evaluation of early oesophageal adenocarcinoma'. In: *Gut* 68.7, pp. 1143–1145 (cit. on p. 80).

Ebigbo, Alanna, Christoph Palm et al. (2019). 'A technical review of artificial intelligence as applied to gastrointestinal endoscopy: clarifying the terminology'. In: *Endoscopy International Open* 7.12, E1616–E1623 (cit. on p. 4).

El Salvador Gastrointestinal Atals (n.d.). URL: `https://www.gastrointestinalatlas.com/english/english.html` (cit. on pp. 6, 7, 22, 29).

Feichtenhofer, Christoph, Axel Pinz and Andrew Zisserman (2016). 'Convolutional two-stream network fusion for video action recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1933–1941 (cit. on pp. 95, 100).

Flejou, JF (2005). 'Barrett's oesophagus: from metaplasia to dysplasia and cancer'. In: *Gut* 54.suppl 1, pp. i6–i12 (cit. on p. 14).

Fogel, Itzhak and Dov Sagi (1989). 'Gabor filters as texture discriminator'. In: *Biological cybernetics* 61.2, pp. 103–113 (cit. on p. 94).

Garcia-Lamont, Farid et al. (2018). 'Segmentation of images by color features: A survey'. In: *Neurocomputing* 292, pp. 1–27 (cit. on p. 32).

Ghatwary, Noha (2017). 'Automatic grade classification of Barretts esophagus through feature enhancement'. In: *Medical Imaging 2017: Computer-Aided Diagnosis*. Vol. 10134. International Society for Optics and Photonics, p. 1013433 (cit. on pp. 59, 61, 63).

Ghatwary, Noha, Amr Ahmed and Xujiong Ye (2017). 'Automated Detection of Barrett's Esophagus Using Endoscopic Images: A Survey'. In: *Annual Conference on Medical Image Understanding and Analysis*. Springer, pp. 897–908 (cit. on p. 4).

Ghatwary, Noha, Xujiong Ye and Massoud Zolgharni (2019). 'Esophageal abnormality detection using DenseNet based Faster R-CNN with Gabor features'. In: *IEEE Access* 7, pp. 84374–84385 (cit. on p. 129).

Ghatwary, Noha, Massoud Zolgharni and Xujiong Ye (2019a). 'Early esophageal adenocarcinoma detection using deep learning methods'. In: *International journal of computer assisted radiology and surgery* 14.4, pp. 611–621 (cit. on p. 129).

– (2019b). 'GFD Faster R-CNN: Gabor Fractal DenseNet Faster R-CNN for Automatic Detection of Esophageal Abnormalities in Endoscopic Images'. In: *International Workshop on Machine Learning in Medical Imaging*. Springer, pp. 89–97 (cit. on p. 129).

Gill, Raghubinder Singh and Rajvinder Singh (2012). 'Endoscopic imaging in Barrett's esophagus: current practice and future applications'. In: *Annals of gastroenterology* 25.2, p. 89 (cit. on p. 17).

Girshick, Ross (2015). 'Fast r-cnn'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448 (cit. on pp. 82, 83).

Girshick, Ross et al. (2015). 'Region-based convolutional networks for accurate object detection and segmentation'. In: *IEEE transactions on pattern analysis and machine intelligence* 38.1, pp. 142–158 (cit. on p. 82).

Goetz, Martin (2012). 'Confocal laser endomicroscopy: applications in clinical and translational science—a comprehensive review'. In: *ISRN Pathology* 2012 (cit. on p. 3).

Goldblum, John R (2003). 'Barrett's esophagus and Barrett's-related dysplasia'. In: *Modern Pathology* 16.4, p. 316 (cit. on p. 3).

Gotoda, Takuji (2007). 'Endoscopic resection of early gastric cancer'. In: *Gastric cancer* 10.1, pp. 1–11 (cit. on p. 16).

Greenspan, Hayit, Bram Van Ginneken and Ronald M Summers (2016). 'Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique'. In: *IEEE Transactions on Medical Imaging* 35.5, pp. 1153–1159 (cit. on pp. 66, 69).

Grisan, Enrico, and Giorgio Diamantis Elisa Veronese et al. (n.d.). 'Computer aided diagnosis of Barrett's esophagus and associated neoplasia using confocal laser endomicroscopy'. In: *Digestive and Liver Disease* (), S147–S148 (cit. on pp. 7, 30, 38, 61–63).

Grisan, Enrico, Elisa Veronese et al. (2012). '239 Computer Aided Diagnosis of Barrett's Esophagus Using Confocal Laser Endomicroscopy: Preliminary Data'. In: *Gastrointestinal Endoscopy* 75.4, AB126 (cit. on p. 37).

Groof, Jeroen de et al. (2019). 'The Argos project: The development of a computer-aided detection system to improve detection of Barrett's neoplasia on white light endoscopy'. In: *United European gastroenterology journal* 7.4, p. 538 (cit. on p. 4).

Guo, Huang et al. (2012). 'Image denoising using fractional integral'. In: *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*. Vol. 2. IEEE, pp. 107–112 (cit. on p. 44).

Haralick, Robert M (1979). 'Statistical and structural approaches to texture'. In: *Proceedings of the IEEE* 67.5, pp. 786–804 (cit. on pp. 33, 45).

Haringsma, Jelle et al. (2001). 'Autofluorescence endoscopy: feasibility of detection of GI neoplasms unapparent to white light endoscopy with an evolving technology'. In: *Gastrointestinal endoscopy* 53.6, pp. 642–650 (cit. on p. 17).

He, Kaiming et al. (2016). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (cit. on pp. 66, 67, 73, 74, 80).

Hiremath, PS et al. (2003). 'Detection of esophageal Cancer (Necrosis) in the Endoscopic images using color image segmentation'. In: *Proceedings of second National*

Conference on Document Analysis and Recognition (NCDAR-2003), Mandya, India, pp. 417–422 (cit. on p. 4).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long short-term memory'. In: *Neural computation* 9.8, pp. 1735–1780 (cit. on p. 134).

Hong, Jisu et al. (2017). 'Convolutional neural network classifier for distinguishing Barrett's esophagus and neoplasia endomicroscopy images'. In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, pp. 2892–2895 (cit. on pp. 39, 59, 61).

Hosseini, Sepidehsadat, Seok Hee Lee and Nam Ik Cho (2018). 'Feeding hand-crafted features for enhancing the performance of convolutional neural networks'. In: *arXiv preprint arXiv:1801.07848* (cit. on p. 67).

Howard, Andrew G et al. (2017). 'Mobilenets: Efficient convolutional neural networks for mobile vision applications'. In: *arXiv preprint arXiv:1704.04861* (cit. on p. 158).

Huang, Gao et al. (2017). 'Densely connected convolutional networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708 (cit. on pp. 67, 74, 75, 87, 132, 138).

Huang, Jian et al. (2018). 'End-to-end continuous emotion recognition from video using 3D ConvLSTM networks'. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6837–6841 (cit. on p. 132).

Humeau-Heurtier, Anne (2019). 'Texture feature extraction methods: A survey'. In: *IEEE Access* 7, pp. 8975–9000 (cit. on p. 33).

Iakovidis, Dimitris K, Eystratios G Keramidas and Dimitris Maroulis (2008). 'Fuzzy local binary patterns for ultrasound texture characterization'. In: *Image analysis and recognition*. Springer, pp. 750–759 (cit. on p. 50).

Iorio, F de et al. (2006). 'Automatic detection of intestinal juices in wireless capsule video endoscopy'. In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 4. IEEE, pp. 719–722 (cit. on p. 67).

Jalab, Hamid A and Rabha W Ibrahim (2012). 'Denoising algorithm based on generalized fractional integral operator with two parameters'. In: *Discrete Dynamics in Nature and Society* 2012 (cit. on p. 43).

– (2013). 'Texture enhancement based on the Savitzky-Golay fractional differential operator'. In: *Mathematical Problems in Engineering* 2013 (cit. on p. 44).

Janse, Mark HA et al. (2016). 'Early esophageal cancer detection using RF classifiers'. In: *SPIE Medical Imaging*. International Society for Optics and Photonics, pp. 97851D–97851D (cit. on p. 77).

Johns Hopkins Department of Pathology: Barrett's Esophagus (n.d.) (cit. on p. 16).

Juefei-Xu, Felix, Vishnu Naresh Boddeti and Marios Savvides (2017). 'Local binary convolutional neural networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19–28 (cit. on p. 5).

Kage, Andreas et al. (2009). 'Narrow-band imaging for the computer assisted diagnosis in patients with Barrett's esophagus'. In: *SPIE Medical Imaging*. International Society for Optics and Photonics, 72603S–72603S (cit. on p. 78).

Kaise, Mitsuru (2015). 'Advanced endoscopic imaging for early gastric cancer'. In: *Best Practice & Research Clinical Gastroenterology* 29.4, pp. 575–587 (cit. on p. 2).

Kamangar, Farin et al. (2009). 'Environmental causes of esophageal cancer'. In: *Gastroenterology Clinics of North America* 38.1, pp. 27–57 (cit. on p. 1).

Kaplan, Lance M (1999). 'Extended fractal analysis for texture classification and segmentation'. In: *IEEE Transactions on Image Processing* 8.11, pp. 1572–1585 (cit. on p. 34).

Kara, MA et al. (2005). 'High-resolution endoscopy plus chromoendoscopy or narrow-band imaging in Barrett's esophagus: a prospective randomized crossover study'. In: *Endoscopy* 37.10, pp. 929–936 (cit. on p. 17).

Karpathy, Andrej, Justin Johnson and Li Fei-Fei (2015). 'Visualizing and understanding recurrent networks'. In: *arXiv preprint arXiv:1506.02078* (cit. on pp. 134, 141).

Kiesslich, Ralf, Martin Goetz et al. (2005). 'Confocal laser endomicroscopy'. In: *Gastrointestinal endoscopy clinics of North America* 15.4, pp. 715–731 (cit. on p. 18).

Kiesslich, Ralf, Liebwin Gossner et al. (2006). 'In vivo histology of Barrett's esophagus and associated neoplasia by confocal laser endomicroscopy'. In: *Clinical Gastroenterology and Hepatology* 4.8, pp. 979–987 (cit. on pp. 15, 63).

Kingsbury, Nick (1999). 'Image processing with complex wavelets'. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 357.1760, pp. 2543–2560 (cit. on p. 42).

Klomp, Sander et al. (2017). 'Evaluation of image features and classification methods for Barrett's cancer detection using VLE imaging'. In: *Medical Imaging 2017: Computer-Aided Diagnosis*. Vol. 10134. International Society for Optics and Photonics, p. 101340D (cit. on p. 79).

Kong, Tao et al. (2016). 'Hypernet: Towards accurate region proposal generation and joint object detection'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 845–853 (cit. on p. 25).

Kovesi, Peter et al. (1999). 'Image features from phase congruency'. In: *Videre: Journal of computer vision research* 1.3, pp. 1–26 (cit. on p. 35).

Krähenbühl, Philipp and Vladlen Koltun (2011). 'Efficient inference in fully connected crfs with gaussian edge potentials'. In: *Advances in neural information processing systems*, pp. 109–117 (cit. on pp. 146, 147).

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'Imagenet classification with deep convolutional neural networks'. In: *Advances in neural information processing systems*, pp. 1097–1105 (cit. on pp. 67, 72, 73, 81).

Kwolek, Bogdan (2005). 'Face detection using convolutional neural networks and Gabor filters'. In: *International Conference on Artificial Neural Networks*. Springer, pp. 551–556 (cit. on pp. 67, 99).

Kwon, Richard S et al. (2009). 'High-resolution and high-magnification endoscopes'. In: *Gastrointestinal endoscopy* 69.3, pp. 399–407 (cit. on p. 17).

Large Scale Visual Recognition Challenge 2014 (2014). URL: `http://image-net.org/challenges/LSVRC/2014/` (cit. on p. 72).

LeCun, Yann et al. (1998). 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324 (cit. on p. 72).

Lee, Michelle H et al. (2012). 'Advanced endoscopic imaging for Barrett's Esophagus: current options and future directions'. In: *Current gastroenterology reports* 14.3, pp. 216–225 (cit. on p. 20).

Li, Mingzhong and Zhaozheng Yin (2016). 'Cell segmentation using stable extremal regions in multi-exposure microscopy images'. In: *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*. IEEE, pp. 526–530 (cit. on p. 48).

Liaw, Andy, Matthew Wiener et al. (2002). 'Classification and regression by randomForest'. In: *R news* 2.3, pp. 18–22 (cit. on p. 36).

Liedlgruber, Michael and Andreas Uhl (2011). 'Computer-aided decision support systems for endoscopy in the gastrointestinal tract: a review'. In: *Biomedical Engineering, IEEE Reviews in* 4, pp. 73–88 (cit. on pp. 16, 18).

Lim, Lee Guan et al. (2011). 'Experienced versus inexperienced confocal endoscopists in the diagnosis of gastric adenocarcinoma and intestinal metaplasia on confocal images'. In: *Gastrointestinal endoscopy* 73.6, pp. 1141–1147 (cit. on p. 3).

Litjens, Geert et al. (2017). 'A survey on deep learning in medical image analysis'. In: *Medical image analysis* 42, pp. 60–88 (cit. on pp. 5, 68).

Liu, Jin et al. (2018). 'Applications of deep learning to MRI images: A survey'. In: *Big Data Mining and Analytics* 1.1, pp. 1–18 (cit. on p. 68).

Liu, Julia, Aldona Dlugosz and Helmut Neumann (2013). 'Beyond white light endoscopy: the role of optical biopsy in inflammatory bowel disease'. In: *World Journal of Gastroenterology: WJG* 19.43, p. 7544 (cit. on p. 18).

Liu, Weibo et al. (2017). 'A survey of deep neural network architectures and their applications'. In: *Neurocomputing* 234, pp. 11–26 (cit. on p. 69).

Liu, Wei et al. (2016). 'Ssd: Single shot multibox detector'. In: *European conference on computer vision*. Springer, pp. 21–37 (cit. on pp. 82, 85, 86, 158).

Liu, Wenqi and Kun Zeng (2018). 'SparseNet: A Sparse DenseNet for Image Classification'. In: *arXiv preprint arXiv:1804.05340* (cit. on p. 67).

Liu, Yijing et al. (2018). 'Dense Convolutional Binary-Tree Networks for Lung Nodule Classification'. In: *IEEE Access* 6, pp. 49080–49088 (cit. on p. 87).

Lowe, G (2004). 'SIFT-The Scale Invariant Feature Transform'. In: *Int. J* 2, pp. 91–110 (cit. on p. 78).

Luan, Shangzhen et al. (2018). 'Gabor convolutional networks'. In: *IEEE Transactions on Image Processing* 27.9, pp. 4357–4366 (cit. on pp. 67, 99).

Manjula, G. N. and Muzameel Ahmed (2017). '2D Shape Representation and Analysis Using Edge Histogram and Shape Feature'. In: *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*. Ed. by Suresh Chandra Satapathy et al. Singapore: Springer Singapore, pp. 545–550. ISBN: 978-981-10-3156-4 (cit. on p. 34).

Matas, Jiri et al. (2004). 'Robust wide-baseline stereo from maximally stable extremal regions'. In: *Image and vision computing* 22.10, pp. 761–767 (cit. on p. 48).

Mathai, Tejas Sudharshan, Vijay Gorantla and John Galeotti (2019). 'Segmentation of Vessels in Ultra High Frequency Ultrasound Sequences Using Contextual Memory'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 173–181 (cit. on p. 132).

Mehta, Rakesh and Karen Egiazarian (2013). 'Rotated Local Binary Pattern (RLBP): Rotation invariant texture descriptor'. In: *2nd International Conference on Pattern Recognition Applications and Methods, ICPRAM 2013, Barcelona, Spain, 15.-18.2.2013*. International Conference on Pattern Recognition Applications and Methods. Contribution: organisation=sgn,FACT1=1
Portfolio EDEND: 2013-12-29
Publisher name: Institute of Electrical and Electronics Engineers IEEE. Institute of Electrical and Electronics Engineers IEEE, pp. 497–502. ISBN: 978-989856541-9 (cit. on p. 47).

Mendel, Robert et al. (2017). 'Barrett's esophagus analysis using convolutional neural networks'. In: *Bildverarbeitung für die Medizin 2017*. Springer, pp. 80–85 (cit. on pp. 5, 8, 66, 80, 104, 108, 118, 120, 122, 124, 129).

Misawa, Masashi et al. (2018). 'Artificial intelligence-assisted polyp detection for colonoscopy: initial experience'. In: *Gastroenterology* 154.8, pp. 2027–2029 (cit. on p. 131).

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, Omar Fawzi et al. (2017). 'Universal adversarial perturbations'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773 (cit. on p. 130).

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi and Pascal Frossard (2016). 'Deepfool: a simple and accurate method to fool deep neural networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582 (cit. on p. 130).

Münzenmayer, Christian (2006). *Color texture analysis in medical applications*. Der Andere Verlag (cit. on p. 79).

Nakai, Yousuke et al. (2014). 'Confocal laser endomicroscopy in gastrointestinal and pancreatobiliary diseases'. In: *Digestive Endoscopy* 26, pp. 86–94 (cit. on p. 4).

Nardi, Giacomo et al. (2019). 'Texture-based classification of confocal laser endomicroscopy images for Barrett's esophagus surveillance'. In: (cit. on p. 38).

Narodytska, Nina and Shiva Kasiviswanathan (2017). 'Simple black-box adversarial attacks on deep neural networks'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, pp. 1310–1318 (cit. on p. 130).

Naveed, Mariam and Kerry B Dunbar (2016). 'Endoscopic imaging of Barrett's esophagus'. In: *World journal of gastrointestinal endoscopy* 8.5, p. 259 (cit. on p. 17).

Nguyen, Anh, Jason Yosinski and Jeff Clune (2015). 'Deep neural networks are easily fooled: High confidence predictions for unrecognizable images'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436 (cit. on p. 130).

Nwoye, Chinedu Innocent et al. (2019). 'Weakly supervised convolutional LSTM approach for tool tracking in laparoscopic videos'. In: *International journal of computer assisted radiology and surgery* 14.6, pp. 1059–1067 (cit. on p. 132).

Ohura, Ryuji et al. (2016). 'Computer-Aided Diagnosis Method for Detecting Early Esophageal Cancer from Endoscopic Image by Using Dyadic Wavelet Transform and Fractal Dimension'. In: *Information Technology: New Generations*. Springer, pp. 929–938 (cit. on p. 76).

Ojala, Timo, Matti Pietikäinen and Topi Mäenpää (2000). 'Gray scale and rotation invariant texture classification with local binary patterns'. In: *European Conference on Computer Vision*. Springer, pp. 404–420 (cit. on pp. 33, 47).

Olah, Christopher (2015). 'Understanding lstm networks'. In: (cit. on pp. 134, 135).

Olliver, JR et al. (2003). 'Chromoendoscopy with methylene blue and associated DNA damage in Barrett's oesophagus'. In: *The Lancet* 362.9381, pp. 373–374 (cit. on p. 19).

Patel, Mitisha Narottambhai and Purvi Tandel (2016). 'A survey on feature extraction techniques for shape based object recognition'. In: *International Journal of Computer Applications* 137.6, pp. 16–20 (cit. on p. 35).

Pereira, Sérgio et al. (2016). 'Brain tumor segmentation using convolutional neural networks in MRI images'. In: *IEEE transactions on medical imaging* 35.5, pp. 1240–1251 (cit. on p. 25).

Pogorelov, Konstantin et al. (2017). 'Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection'. In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, pp. 164–169 (cit. on pp. 6, 7, 22, 29, 126).

Pu, Yi-Fei, Ji-Liu Zhou and Xiao Yuan (2010). 'Fractional differential mask: a fractional differential-based approach for multiscale texture enhancement'. In: *Image Processing, IEEE Transactions on* 19.2, pp. 491–511 (cit. on p. 42).

Putten, Joost van der et al. (2019). 'Informative Frame Classification of Endoscopic Videos Using Convolutional Neural Networks and Hidden Markov Models'. In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 380–384 (cit. on p. 81).

Qadir, Hemin Ali et al. (2019). 'Improving Automatic Polyp Detection Using CNN by Exploiting Temporal Dependency in Colonoscopy Video'. In: *IEEE Journal of Biomedical and Health Informatics* (cit. on pp. 132, 158, 159).

Qi, Xin et al. (2010). 'Image analysis for classification of dysplasia in Barrett's esophagus using endoscopic optical coherence tomography'. In: *Biomedical optics express* 1.3, pp. 825–847 (cit. on p. 19).

Rajan, P et al. (2009). 'Automated diagnosis of Barrett's esophagus with endoscopic images'. In: *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer, pp. 2189–2192 (cit. on pp. 3, 79).

Rajendra, Shanmugarajah and Prateek Sharma (2017). 'Barrett Esophagus and Intramucosal Esophageal Adenocarcinoma'. In: *Hematology/Oncology Clinics of North America* (cit. on p. 14).

Ramirez, Francisco C et al. (2005). 'Feasibility and safety of string, wireless capsule endoscopy in the diagnosis of Barrett's esophagus'. In: *Gastrointestinal endoscopy* 61.6, pp. 741–746 (cit. on p. 19).

Ren, Shaoqing et al. (2015). 'Faster r-cnn: Towards real-time object detection with region proposal networks'. In: *Advances in neural information processing systems*, pp. 91–99 (cit. on pp. 82, 84, 97, 101, 158).

Scheeve, Thom et al. (2019). 'A novel clinical gland feature for detection of early Barrett's neoplasia using volumetric laser endomicroscopy'. In: *Medical Imaging 2019: Computer-Aided Diagnosis*. Vol. 10950. International Society for Optics and Photonics, 109501Y (cit. on p. 79).

Schölvinck, Dirk W et al. (2017). 'Detection of lesions in dysplastic Barrett's esophagus by community and expert endoscopists'. In: *Endoscopy* 49.02, pp. 113–120 (cit. on p. 2).

Sekiguchi, Masau and Ichiro Oda (2017). 'High miss rate for gastric superficial cancers at endoscopy: what is necessary for gastric cancer screening and surveillance using endoscopy?' In: *Endoscopy international open* 5.08, E727–E728 (cit. on p. 3).

Setio, Arnaud A. A. et al. (2013). 'Evaluation and Comparison of Textural Feature Representation for the Detection of Early Stage Cancer in Endoscopy'. In: *Proceedings of the International Conference on Computer Vision Theory and Applications (VISIGRAPP 2013)*, pp. 238–243 (cit. on p. 76).

Shaheen, Nicholas and David F Ransohoff (2002). 'Gastroesophageal reflux, Barrett esophagus, and esophageal cancer: scientific review'. In: *Jama* 287.15, pp. 1972–1981 (cit. on p. 2).

Shahid, Muhammad W and Michael B Wallace (2010). 'Endoscopic imaging for the detection of esophageal dysplasia and carcinoma'. In: *Gastrointestinal endoscopy clinics of North America* 20.1, pp. 11–24 (cit. on p. 19).

Shi, Qiaoqiao et al. (2018). 'Deep CNN With Multi-Scale Rotation Invariance Features for Ship Classification'. In: *Ieee Access* 6, pp. 38656–38668 (cit. on p. 67).

Simonyan, Karen and Andrew Zisserman (2014). 'Very deep convolutional networks for large-scale image recognition'. In: *arXiv preprint arXiv:1409.1556* (cit. on pp. 67, 72, 74, 81, 86).

Singh, Rajvinder and Sze Pheh Yeap (2015). 'Endoscopic imaging in Barrett's esophagus'. In: *Expert review of gastroenterology & hepatology* 9.4, pp. 475–485 (cit. on p. 19).

Sommen, Fons van der et al. (2016). 'Computer-aided detection of early neoplastic lesions in Barrett's esophagus'. In: *Endoscopy* (cit. on pp. 77, 104, 108).

Souza, Luis A de et al. (2018). 'A survey on Barrett's esophagus analysis using machine learning'. In: *Computers in biology and medicine* (cit. on p. 4).

Souza, Luis et al. (2017). 'Barrett's esophagus analysis using SURF features'. In: *Bildverarbeitung für die Medizin 2017*. Springer, pp. 141–146 (cit. on pp. 66, 77).

Struyvenberg, Maarten R et al. (2019). '297–Deep Learning Algorithm for Characterization of Barrett's Neoplasia Demonstrates High Accuracy on Nbi-Zoom Images'. In: *Gastroenterology* 156.6, S–58 (cit. on p. 81).

Su, Jiawei, Danilo Vasconcellos Vargas and Kouichi Sakurai (2019). 'One pixel attack for fooling deep neural networks'. In: *IEEE Transactions on Evolutionary Computation* (cit. on p. 130).

Sub-Challenge Early Barrett's cancer detection (n.d.). URL: `https://endovissub-barrett.grand-challenge.org` (cit. on pp. 6, 7, 21, 29, 126).

Swager, Anne-Fré et al. (2017). 'Computer-aided detection of early Barrett's neoplasia using volumetric laser endomicroscopy'. In: *Gastrointestinal endoscopy* 86.5, pp. 839–846 (cit. on p. 79).

Szegedy, Christian, Sergey Ioffe et al. (2017). 'Inception-v4, inception-resnet and the impact of residual connections on learning'. In: *Thirty-First AAAI Conference on Artificial Intelligence* (cit. on p. 158).

Szegedy, Christian, Wei Liu et al. (2015). 'Going deeper with convolutions'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 (cit. on pp. 73, 81).

Tajbakhsh, Nima, Suryakanth R Gurudu and Jianming Liang (2015). 'Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks'. In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 79–83 (cit. on p. 131).

Tou, Jing Yi, Phooi Yee Lau and Yong Haur Tay (2007). 'Computer vision-based wood recognition system'. In: *Proceedings of International workshop on advanced image technology*. Citeseer (cit. on p. 45).

Trovato, Cristina et al. (2013). 'Confocal laser endomicroscopy for in vivo diagnosis of Barrett's oesophagus and associated neoplasia: a pilot study conducted in a single Italian centre'. In: *Digestive and Liver Disease* 45.5, pp. 396–402 (cit. on p. 2).

Tychsen-Smith, Lachlan and Lars Petersson (2018). 'Improving object localization with fitness nms and bounded iou loss'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6877–6885 (cit. on p. 25).

Uijlings, Jasper RR et al. (2013). 'Selective search for object recognition'. In: *International journal of computer vision* 104.2, pp. 154–171 (cit. on p. 82).

UPPER ENDOSCOPY (n.d.). URL: `https://www.viralpatelmd.com/upper-endoscopy-egd-best-doctor-to-do-egd-in-dallas-rockwall-rowlett-tx/` (cit. on p. 17).

Van Der Sommen, Fons, Fons Zinger Svitlana et al. (2014). 'Supportive automatic annotation of early esophageal cancer using local gabor and color features'. In: *Neurocomputing* 144, pp. 92–106 (cit. on pp. 66, 77, 118, 120, 164).

Van Der Sommen, Fons, Svitlana Zinger, Erik J Schoon et al. (2013). 'Computer-aided detection of early cancer in the esophagus using HD endoscopy images'. In: *SPIE Medical Imaging*. International Society for Optics and Photonics, pp. 86700V–86700V (cit. on p. 77).

Van Riel, Sjors et al. (2018). 'Automatic detection of early esophageal cancer with CNNS using transfer learning'. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 1383–1387 (cit. on pp. 8, 80).

Veronese, Elisa et al. (2013). 'Hybrid patch-based and image-wide classification of confocal laser endomicroscopy images in Barrett's esophagus surveillance'. In: *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*. IEEE, pp. 362–365 (cit. on pp. 7, 15, 30, 38, 61–63).

Visrodia, Kavel et al. (2016). 'Yield of repeat endoscopy in Barrett's esophagus with no dysplasia and low-grade dysplasia: a population-based study'. In: *Digestive diseases and sciences* 61.1, pp. 158–167 (cit. on p. 2).

Wallace, Michael B and Paul Fockens (2009). 'Probe-based confocal laser endomicroscopy'. In: *Gastroenterology* 136.5, pp. 1509–1513 (cit. on p. 18).

Wang, Kenneth K and Richard E Sampliner (2008). 'Updated guidelines 2008 for the diagnosis, surveillance and therapy of Barrett's esophagus'. In: *The American journal of gastroenterology* 103.3, p. 788 (cit. on p. 15).

Wang, Thomas D et al. (2007). 'Functional imaging of colonic mucosa with a fibered confocal microscope for real-time in vivo pathology'. In: *Clinical gastroenterology and hepatology* 5.11, pp. 1300–1305 (cit. on p. 18).

Watson, Thomas J (2014). 'Endoscopic therapies for Barrett's neoplasia'. In: *Journal of thoracic disease* 6.Suppl 3, S298 (cit. on p. 16).

Worldwide cancer data (n.d.). URL: `https://www.wcrf.org/dietandcancer/cancer-trends/worldwide-cancer-data` (cit. on p. 1).

Wu, Po-Cheng and Liang-Gee Chen (2001). 'An efficient architecture for two-dimensional discrete wavelet transform'. In: *IEEE Transactions on circuits and systems for video technology* 11.4, pp. 536–545 (cit. on p. 42).

Xingjian, SHI et al. (2015). 'Convolutional LSTM network: A machine learning approach for precipitation nowcasting'. In: *Advances in neural information processing systems*, pp. 802–810 (cit. on pp. 133, 141, 142).

Xu, Xuanang et al. (2019). 'Efficient Multiple Organ Localization in CT Image using 3D Region Proposal Network'. In: *IEEE transactions on medical imaging* (cit. on p. 143).

Yamaguchi, Jumpei, Akihiko Yoneyama and Teruya Minamoto (2015). 'Automatic detection of early esophageal cancer from endoscope image using fractal dimension and discrete wavelet transform'. In: *Information Technology-New Generations (ITNG), 2015 12th International Conference on*. IEEE, pp. 317–322 (cit. on pp. 75, 76).

Yang, Mingqiang, Kidiyo Kpalma and Joseph Ronsin (2008). *A survey of shape feature extraction techniques* (cit. on p. 34).

Yao, Hu et al. (2016). 'Gabor feature based convolutional neural network for object recognition in natural scene'. In: *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, pp. 386–390 (cit. on pp. 67, 99).

Yi, Jingru et al. (2017). 'Fast neural cell detection using light-weight SSD neural network'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 108–112 (cit. on p. 66).

Yousefi, Mahboubeh sadat et al. (2018). 'Esophageal cancer in the world: incidence, mortality and risk factors'. In: *Biomedical Research and Therapy* 5.7, pp. 2504–2517 (cit. on p. 1).

Youssef, Sherin M, Ahmed Abou ElFarag and Noha M Ghatwary (2014). 'Adaptive video watermarking integrating a fuzzy wavelet-based human visual system perceptual model'. In: *Multimedia Tools and Applications* 73.3, pp. 1545–1573 (cit. on pp. 42, 50).

Yu, Fisher et al. (2018). 'Deep layer aggregation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412 (cit. on p. 143).

Yu, Lequan et al. (2016). 'Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos'. In: *IEEE journal of biomedical and health informatics* 21.1, pp. 65–75 (cit. on p. 131).

Yu, Qiang et al. (2013). 'The use of a Riesz fractional differential-based approach for texture enhancement in image processing'. In: *ANZIAM Journal* 54, pp. 590–607 (cit. on p. 43).

Zhang, Ruikai et al. (2018). 'Polyp detection during colonoscopy using a regression-based convolutional neural network with a tracker'. In: *Pattern recognition* 83, pp. 209–219 (cit. on p. 132).

Zhao, Yuan-Yuan et al. (2019). 'Computer-assisted diagnosis of early esophageal squamous cell carcinoma using narrow-band imaging magnifying endoscopy'. In: *Endoscopy* 51.04, pp. 333–341 (cit. on p. 4).

Zhu, Guangming et al. (2018). 'Continuous Gesture Segmentation and Recognition Using 3DCNN and Convolutional LSTM'. In: *IEEE Transactions on Multimedia* 21.4, pp. 1011–1021 (cit. on p. 132).

Zhu, Hongyuan, Romain Vial and Shijian Lu (2017). 'Tornado: A spatio-temporal convolutional regression network for video action proposal'. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5813–5821 (cit. on p. 132).